



Week 7_예습과제_김정은

6.1 이미지 분류를 위한 신경망

- 입력 데이터로 이미지를 사용한 분류 → 특정 대상이 영상 내에 존재하는지 여부를 판단

LeNet-5

- LeNet-5 는 합성곱과 다운 샘플링 혹은 풀링을 반복적으로 거치면서 마지막에 완전연결층에서 분류를 수행
- `transforms.Compose` : 이미지를 변형할 수 있는 방식들의 묶음
- `transforms.RandomResizedCrop` : 입력 이미지를 주어진 크기로 조정. 또한 scale 은 원래 이미지를 임의의 크기 만큼 면적을 무작위로 자르겠다는 의미
- `transforms.RandomHorizontalFlip` : 주어진 확률로 이미지를 수평 반전 시킴. 확률 값을 지정하지 않았으므로 기본값인 0.5의 확률로 이미지들이 수평 반전됨. 즉, 훈련 이미지 중 반은 위아래 뒤집힌 상태로 두고 반은 그대로 사용
- `transforms.ToTensor` : ImageFolder 메서드를 비롯해 torchvision 메서드는 이미지를 읽을 때 파이썬 이미지 라이브러리인 PIL을 사용. PIL을 사용해 이미지를 읽으면 생성되는 이미지는 범위가 [0, 255]이며, 배열의 차원이 (높이 H X 너비 W X 채널 수 C)로 표현됨. 이후 효율적인 연산을 위해 torch.FloatTensor 배열로 바꾸어야 하는데, 이때 픽셀 값의 범위는 [0, 0, 1, 0] 사이가 되고 차원의 순서도 (채널 수 C X 높이 H X 너비 W)로 바뀜. 이러한 작업을 수행해주는 메서드가 ToTensor()
- `transforms.Normalize` : 전이 학습에서 사용하는 사전 훈련된 모델들은 대개 ImageNet 데이터셋에서 훈련되었음. 따라서 사전 훈련된 모델을 사용하기 위해서는 ImageNet 데이터의 각 채널별 평균과 표준편차에 맞는 정규화(Normalize)를 해주어야 함. Normalize 메서드 안에 사용된 mean과 std는 ImageNet 에서 이미지들의 RGB 채널마다 평균과 표준편차를 의미.

- `__call__` : 클래스를 호출할 수 있도록 하는 메서드. `__init__`은 인스턴스 초기화를 위해 사용한다면 `__call__`은 인스턴스가 호출되었을 때 실행. 클래스에 이 함수가 있을 경우, 클래스 객체 자체를 호출하면 이 함수의 리턴 값이 반환됨
- 넘파이 `random()` 함수는 임의의 난수를 생성하는데, 이때 난수를 생성하기 위해 사용되는 것이 시드 값(seed value). 또한, `Numpy.random.seed()` 메서드는 상태를 초기화시킴. 즉, 이 모듈이 호출될 때마다 임의의 난수가 재생성된다. 하지만 특정 시드 값을 부여하면 상태가 저장되기 때문에 동일한 난수를 생성.
- `cv2.cvtColor` : 이미지의 색상을 변경하기 위해 사용되며 파라미터는 다음과 같다.
`cv2.cvtColor(image, cv2.COLOR_BGR2RGB)`
 - 첫 번째 파라미터 : 입력 이미지
 - 두 번째 파라미터 : 변환할 이미지의 색상을 지정하는 것으로 BGR(Blue, Green, Red) 채널 이미지를 RGB(컬러)로 변경하겠다는 의미
- 이 경우, 데이터 셋 크기가 클 수 있으므로 `__init__`에서 전체 데이터를 읽어 오는 것이 아니라 경로만 저장해 놓고, `__getitem__` 메서드에서 이미지를 읽어온다. 즉, 데이터를 어디에서 가져올지 결정.
- `DogvsCatDataset()` 클래스 : 데이터를 불러오는 방법을 정의
- 파이토치의 데이터로더는 배치 관리를 담당. 한 번에 모든 데이터를 불러오면 메모리에 부담을 줄 수 있으므로 데이터를 그룹으로 쪼개서 조금씩 불러온다.
- `batch_size` : 한 번에 메모리로 불러올 데이터 크기
- `shuffle` : 메모리로 데이터를 가져올 때 임의로 섞어서 가져오도록 한다.
- `summary` : 모델의 네트워크 관련 정보를 확인할 수 있다.
- 모멘텀 SGD : 경사하강법으로 SGD에 관성이 추가되었다. 매번 기울기를 구하지만 가중치를 수정하기 전에 이전 수정 방향을 참고하여 같은 방향으로 일정한 비율만 수정되게 하는 방법이다.
- `torch.unsqueeze` : 텐서에 차원을 추가할 때 사용. 또한, (0)은 차원이 추가될 위치를 의미
- 소프트맥스(softmax) : 지정된 차원을 따라 텐서의 요소(텐서의 개별 값)가 (0, 1) 범위에 있고 합계가 1이 되도록 크기를 다시 조정
- 결론적으로, 예측 결과 레이블이 0.5보다 크면 개를 의미, 0.5보다 작으면 고양이를 의미

6.1.2 AlexNet

- 합성곱층 총 다섯 개와 완전연결층 세 개로 구성
- 맨 마지막 완전연결층은 카테고리 1000개를 분류하기 위해 softmax 활성화 함수를 사용
- 전체적으로 보면 GPU 두 개를 기반으로 한 병렬 구조인 점을 제외하면 LeNet -5 와 크게 다르지 않음
- AlexNet 의 합성곱층에서 사용된 활성화 함수는 렐루(ReLU)
- ReLU 활성화 함수에서 inplace : 연산에 대한 결과값을 새로운 변수에 저장하는 것이 아닌 기존 데이터를 대체하는 것. 즉, 기존 값을 연산 결과값으로 대체함으로써 기존 값들을 무시하겠다는 의미.