

5장 합성곱 신경망2

📅 날짜	@2025년 3월 26일 → 2025년 3월 31일
📌 선택	DL세션 과제
📎 실습 파일	<u>Week4_예습과제_이수나 (1).ipynb</u>
📌 주차	4주차
🌟 진행 상태	완료

5.3 전이 학습

5.3.1 특성 추출 (feature extractor)

특성 추출 기법 실습

(Note) 계산 그래프

5.3.2 미세 조정(fine-tuning) 기법

5.4 설명 가능한 CNN(explainable CNN)

5.4.1 특성 맵 시각화 실습

5.5 그래프 합성곱 네트워크

5.5.1 그래프

5.5.2 그래프 신경망

5.5.3 그래프 합성곱 네트워크

5.3 전이 학습

: 이미지넷(ImageNet)처럼 아주 큰 데이터셋을 써서 훈련된 모델의 가중치를 가져와 해결하려는 과제에 맞게 보정해서 사용하는 것

아주 큰 데이터셋을 사용하여 훈련된 모델: 사전 훈련된 모델(네트워크)

[문제 A] - 해결 → [문제 A를 해결하면서 얻은 지식 및 경험] -적용(전이학습) → [문제B]

5.3.1 특성 추출 (feature extractor)

: ImageNet 데이터셋으로 사전 훈련된 모델을 가져온 후, 마지막에 완전연결층 부분만 새로 만듦.

→ 학습할 때는 마지막 완전연결층만 학습하고, 나머지 계층들은 학습되지 않도록 함.

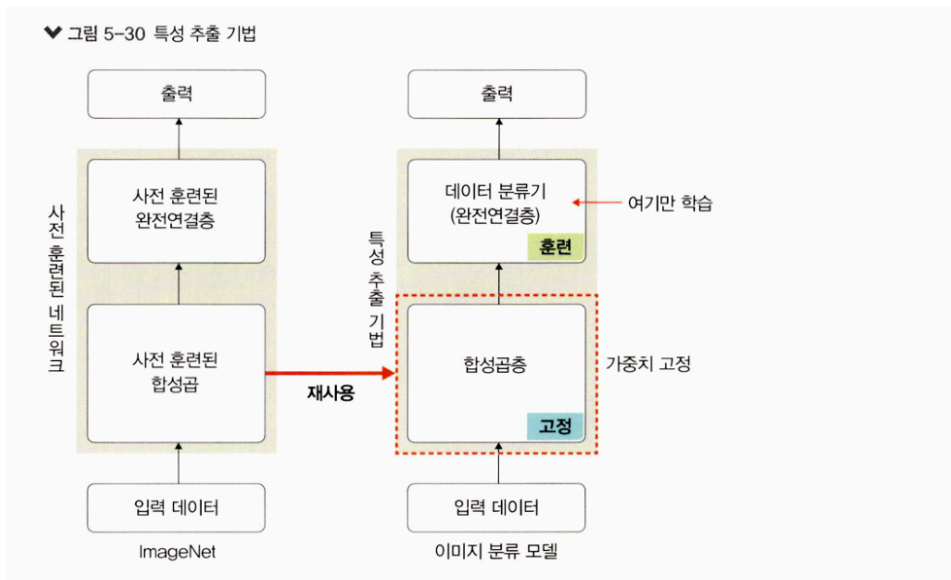
- 특성 추출은 이미지 분류를 위해 두 부분으로 구성됨



- **합성곱층**: 합성곱층 & 풀링층
- **데이터 분류기(완전연결층)**: 추출된 특성을 입력받아 최종적으로 이미지에 대한 클래스를 분류하는 부분

사전 훈련된 네트워크의 합성곱층(가중치 고정)에 새로운 데이터를 통과시키고, 그 출력을 데이터 분류기에서 훈련 시킴

- Xception, Inception V3, ResNet50, VGG16, VGG19, MobileNet



특성 추출 기법 실습

- RandomResizedCrop으로 이미지를 랜덤한 비율로 자른 후 데이터 크기를 조정
⇒ 데이터 확장 용도

```
pip install mxnet
pip instal --user mxnet # 설치 실패 시

import matplotlib.pyplot as plt
import mxnet as mx
from mxnet.gluon.data.vision import transforms
```

```

example_image = mx.image.imread("cat.jpg")
plt.imshow(example_image.asnumpy())

# 함수 생성
def show_images(imgs, num_rows, num_cols, scale=2):
    aspect_ratio = imgs[0].shape[0]/imgs[0].shape[1]
    figsize = (num_cols*scale, num_rows*scale*aspect_ratio)
    _, axes = plt.subplots(num_rows, num_cols, figsize=figsize)

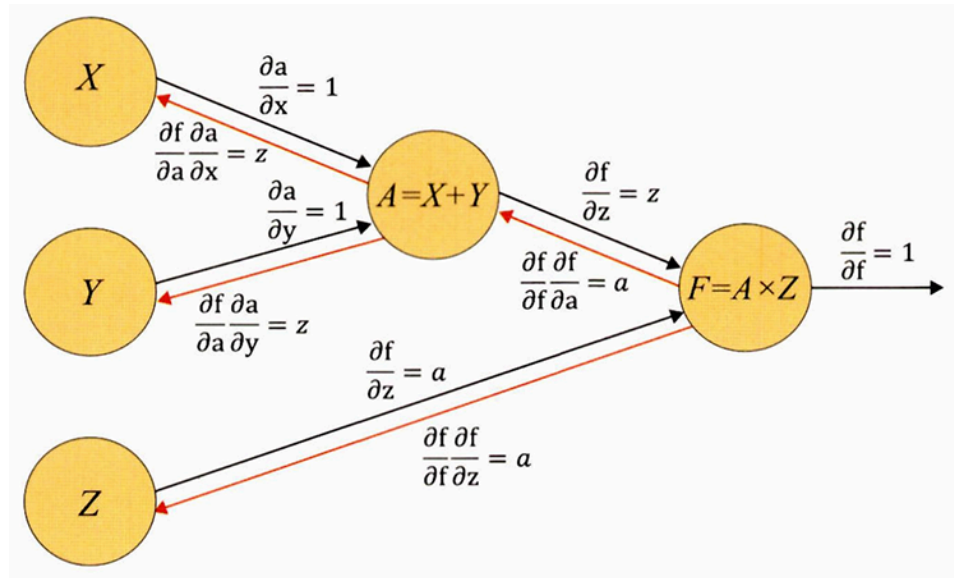
    for i in range(num_rows):
        for j in range(num_cols):
            axes[i][j].imshow(imgs[i*num_cols+j].asnumpy())
            axes[i][j].axes.get_xaxis().set_visible(False) #x축 전체를 숨김
            axes[i][j].axes.get_yaxis().set_visible(False) #y축 전체를 숨김
    plt.subplots_adjust(hspace=0.1, wspace=0)
    return axes

def apply(img, aug, num_rows=2, num_cols=4, scale=3):
    Y = [aug(img) for _ in range(num_rows*num_cols)]
    show_images(Y, num_rows, num_cols, scale)
    # 다양한 샘플을 얻기 위해 여러 번 데이터 확장 적용

# 적용된 이미지 출력
shape_aug = transforms.RandomResizedCrop(size=(200,100), scale=(0.1,1), ra
# 출력할 크기를 200*200으로 조정
# 면적 비율을 0.1~1 범위내에서 무작위로 자름
# 면적의 너비와 높이 비율을 0.5~2 범위 내에서 무작위로 조정
apply(example_image, shape_aug)

```

(Note) 계산 그래프



: 계산 과정을 그래프로 나타냄, node와 edge로 구성

- 국소적 계산이 가능
 - 그림에서 Z값이 변경되었다면, X,Y 계산 결과를 유지한 채 연산이 필요한 $F=A*Z$ 만 계산
- 역전파를 통한 미분 계산이 편리 by chain rule

구분	메모리	계산 그래프 상주 유무
tensor.clone()	새롭게 할당	계속 상주
tensor.detach()	공유해서 사용	상주하지 않음
tensor.clone().detach()	새롭게 할당	상주하지 않음

- tensor.clone()은 기본 텐서의 내용을 복사한 텐서 생성
- detach()는 기존 텐서에서 기울기가 전파되지 않는 텐서
- tensor.clone().detach()는 기존 텐서를 복사한 새로운 텐서를 생성하지만, 기울기에 영향을 주지 X

5.3.2 미세 조정(fine-tuning) 기법

: 특성 추출 기법에서 더 나아가, 사전 훈련된 모델과 합성곱층, 데이터 분류기의 가중치를 업데이트하여 훈련시키는 방법

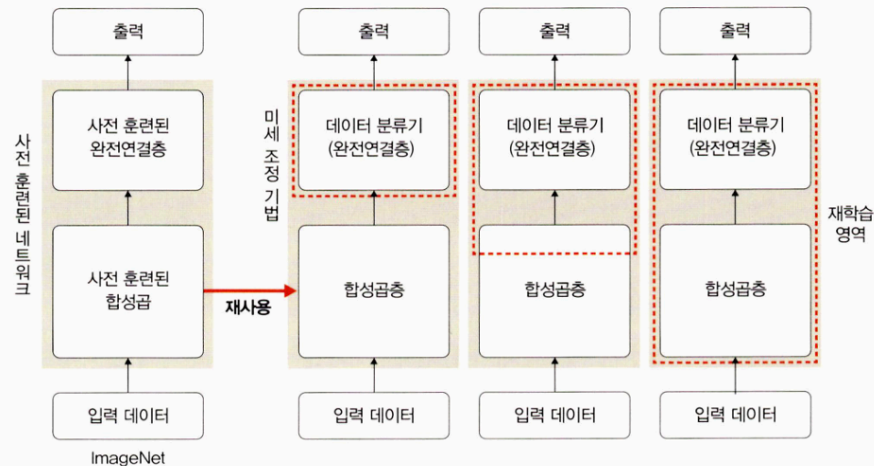
- 특성이 잘못 추출되었다면 미세 조정 기법으로 새로운 이미지 데이터를 사용하여 네트워크의 가중치를 업데이트해서 특성을 다시 추출

사전 학습된 모델을 목적에 맞게 재학습시키거나 학습된 가중치의 일부를 재학습 시키는 것

미세 조정 기법은 사전 훈련된 네트워크를 미세 조정하여 분석하려는 데이터셋에 잘 맞도록 모델의 파라미터를 조정하는 기법

	데이터셋	사전 훈련된 모델과 유사성	전략
1	큼	작음	모델 전체를 재학습시킴
2	큼	큼	합성곱층의 뒷부분과 데이터 분류기를 학습시킴 (데이터셋이 유사하기 때문에 전체를 학습시키는 것보다는 강한 특징이 나타나는 합성곱층의 뒷부분과 데이터 분류기를 학습)
3	작음	작음	합성곱층의 일부분과 데이터 분류기를 학습 (데이터가 적어 일부 계층에 미세 조정 기법을 적용하는 것이 효과가 없을 수 있음 → 합성곱층 중 어디까지 새로 학습시켜야 할지 적당히 설정해야)
4	작음	큼	데이터 분류기만 학습, 데이터가 적기 때문에 많은 계층에 미세 조정 기법을 적용하여 과적합이 발생할 수 있음. ⇒ 완전연결층에 대해서만 미세 조정 기법을 적용

▼ 그림 5-44 미세 조정 기법



5.4 설명 가능한 CNN(explainable CNN)

: 딥러닝 처리 결과를 사람이 이해할 수 있는 방식으로 제시

→ CNN 처리 과정을 시각화 → 결과에 대한 신뢰성 획득

[입력 x] → [블랙박스 blackbox] → [출력 y]

5.4.1 특성 맵 시각화 실습

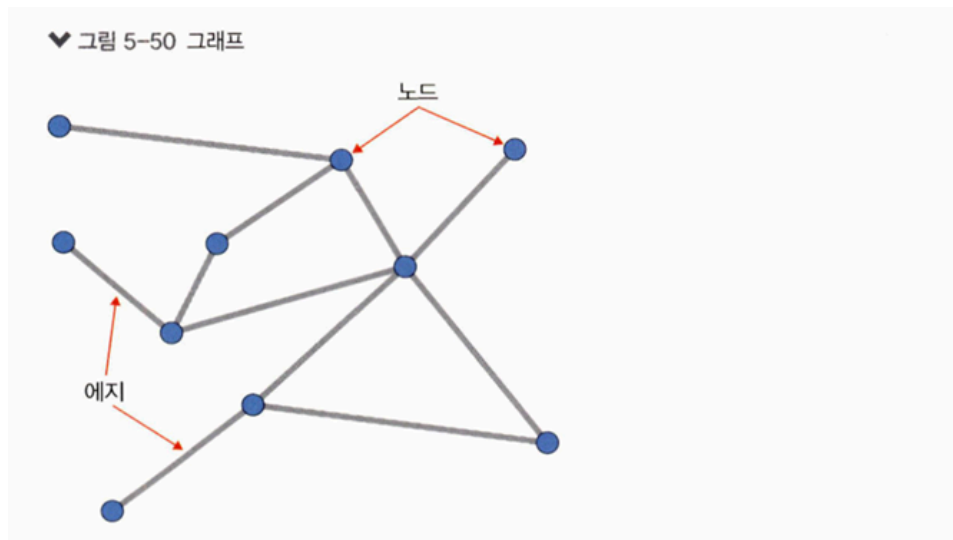


특성맵: 입력 이미지 또는 다른 특성 맵처럼 필터를 입력에 적용한 결과 → 특성 맵을 시각화하여 입력 특성을 감지하는 방법을 이해하도록 도움

5.5 그래프 합성곱 네트워크

: 그래프 데이터를 위한 신경망(graph convolutional network)

5.5.1 그래프



방향성이 있거나(directed) 없는(undirected) edge로 연결된 노드(nodes=vertices)의 집합

- 노드는 원소들을 의미하고, edge는 결합 방법(single, double, triple, aromatic 등)을 의미함

5.5.2 그래프 신경망

: 그래프 구조에서 사용하는 신경망

(1단계) 인접행렬(adjacency matrix)

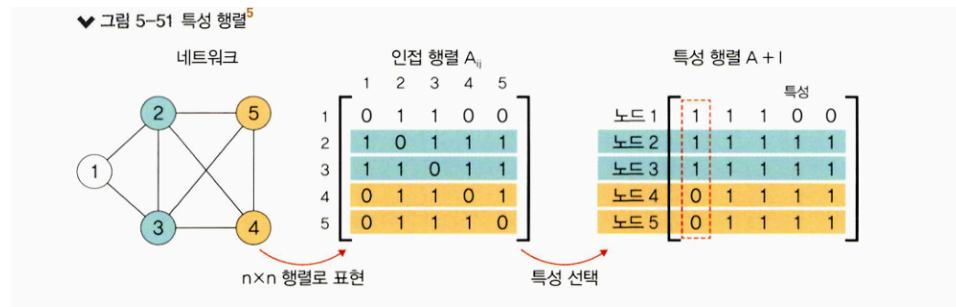
- 노드 n 개를 $n \times n$ 행렬로 표현
- 생성된 인접 행렬 내의 값은 'A_{ij}는 i와 j의 관련성 여부'를 만족하는 값으로 채워줌

⇒ 인접 행렬 과정은 컴퓨터가 이해하기 쉽게 그래프로 표현하는 과정

(2단계) 특성행렬(feature matrix)

- 인접 행렬만으로는 특성을 파악하기 어려워 단위 행렬 적용
- 각 입력 데이터에서 이용할 특성을 선택함
- 특성 행렬에서 각 행은 선택된 특성에 대해 각 노드가 갖는 값을 의미

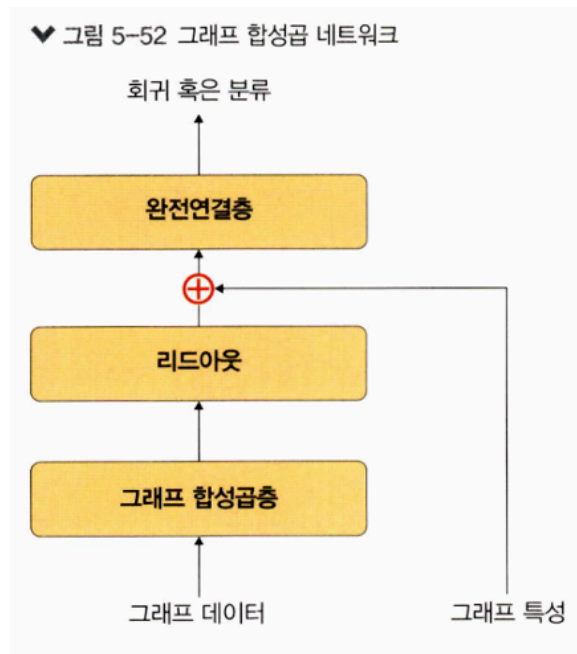
▼ 그림 5-51 특성 행렬⁵



5.5.3 그래프 합성곱 네트워크

: 이미지에 대한 합성곱을 그래프 데이터로 확장한 알고리즘

(Graph Convolutional network, GCN)



- 리드아웃(readout): 특성 행렬을 하나의 벡터로 변환하는 함수

⇒ 전체 노드의 특성 벡터에 대해 평균을 구하고 그래프 전체를 표현하는 하나의 벡터를 생성

GCN에서 가장 중요한 부분은 그래프 합성곱층

그래프 합성곱층을 이용한 그래프 형태의 데이터는 행렬 데이터로 변환되어 딥러닝 알고리즘을 적용할 수 있기 때문.

GCN 활용

- SNS에서 관계 네트워크
- 학술 연구에서 인용 네트워크
- 3D Mesh