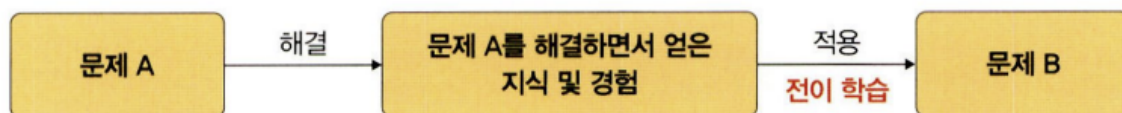




# Ch.5 합성곱 신경망 1

## 5.3 전이 학습

- 합성곱 신경망 기반 딥러닝 모델 훈련 → 방대한 양의 데이터 필요 → 많은 돈/시간 필요
  - 전이 학습(transfer learning)** : 큰 데이터셋(이미지넷) 사용 → 훈련된 모델의 가중치를 보정하여 사용
    - 사전 훈련된 모델(네트워크)** : 큰 데이터셋을 이용하여 훈련된 모델
- ⇒ 비교적 적은 수의 데이터로 과제 해결 가능!



### 5.3.1 특성 추출 기법

- **특성 추출(feature extractor)** : ImageNet 데이터셋으로 사전 훈련된 모델 불러오기  
→ 마지막 완전연결층 부분만 새로 생성

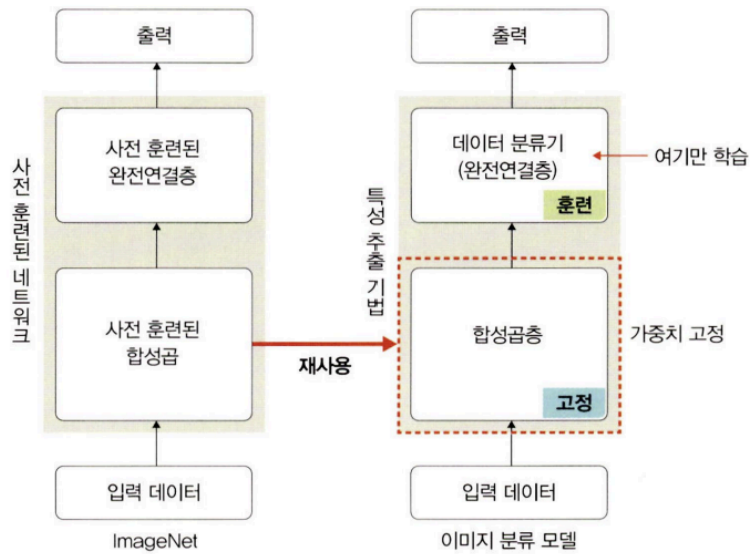
→ 학습 시 마지막 완전연결층(이미지의 카테고리 결정하는 역할)만 학습, 다른 부분은 학습 X

1. **합성곱층** : 합성곱층 & 풀링층으로 구성
2. **데이터 분류기(완전연결층)** : 추출된 특성을 입력받아 최종적으로 이미지에 대한 클래스를 분류하는 부분

- 사전 훈련된 네트워크의 합성곱층(가중치 고정)에 새로운 데이터 통과, 그 출력을 이미지 분류기에서 훈련

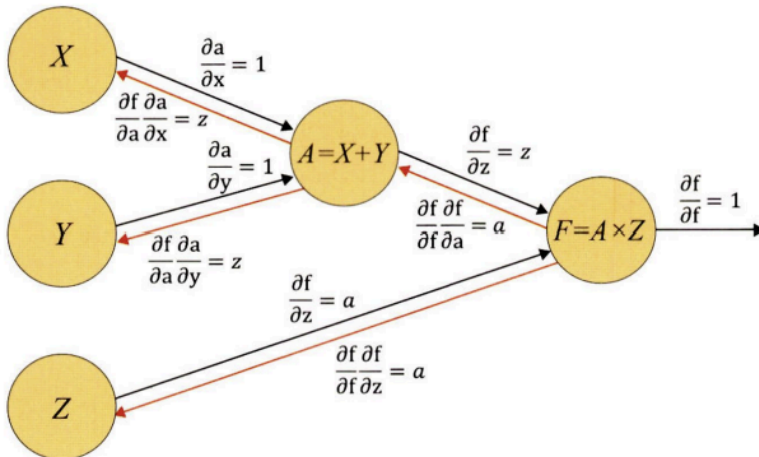
### [사용 가능한 이미지 분류 모델]

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



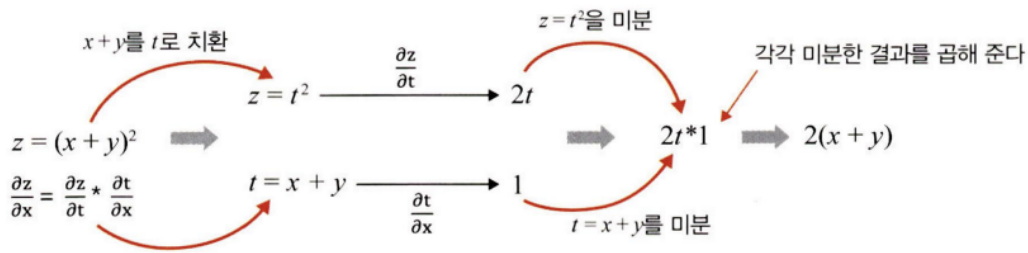
### [계산 그래프(Computational graph)]

- 계산 과정을 그래프로 나타낸 것
- 여러 개의 노드(node) & 연결선(edge)로 구성
- 국소적 계산 가능
- 역전파를 통한 미분 계산이 편리(연쇄법칙(chain rule) 사용)



### [연쇄 법칙(chain rule)]

- 두 개 이상의 함수가 결합된 함수
- 각 합성 함수의 미분 결과를 곱한 결과로 나타내는 것



### 5.3.2 미세 조정 기법(fine-tuning)

- 사전 훈련된 모델과 합성곱층, 데이터 분류기의 가중치를 업데이트하여 훈련시키는 방식
- 특성 추출 - 목표 특성을 잘 추출했다는 전제 하에 좋은 성능 도출 가능
- 특성 추출 Bad → **미세 조정 기법**으로 새로운 이미지 데이터 사용 → 네트워크 가중치 업데이트 → 특성 재추출

⇒ 사전 학습된 모델을 목적에 맞게 재학습 or 학습된 가중치의 일부를 재학습

#### [미세 조정 기법 전략]

##### 1. 데이터셋 Big & 사전 훈련된 모델과의 유사성 Small

- 데이터 크기 Big → 모델 전체 재학습

##### 2. 데이터셋 Big & 사전 훈련된 모델과의 유사성 Big

- 합성곱층의 뒷부분 & 데이터 분류기 학습
- 데이터셋 유사 → 전체 학습보다 합성곱층의 뒷부분 & 데이터 분류기(강한 특성) 학습하는 것이 유리

##### 3. 데이터셋 Small & 사전 훈련된 모델과의 유사성 Small

- 합성곱층의 일부분 & 데이터 분류기 학습
- 데이터 적음 → 미세 조정 기법 효과 X 수 있음!

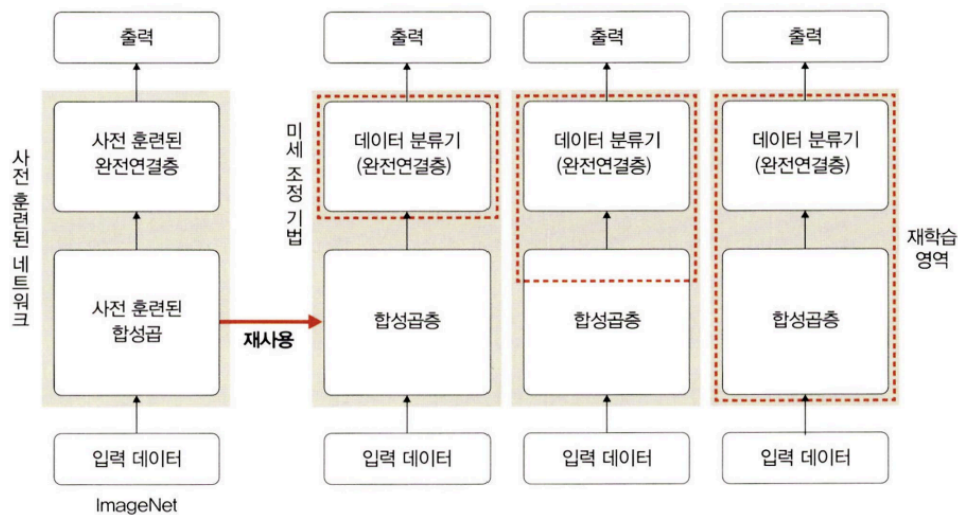
⇒ 합성곱층 학습 부분 선택 중요!

#### 4. 데이터셋 Small & 사전 훈련된 모델과의 유사성 Big

- 데이터 분류기만 학습
- 데이터 적음 → 많은 계층에 미세 조정 기법 적용 시 과적합 가능성 O

⇒ 최종 데이터 분류기인 완전연결층에만 미세 조정 기법 적용!

★ 정교하고 미세한 파라미터 업데이트 중요!



## 5.4 설명 가능한 CNN(explainable CNN)

- 딥러닝 처리 결과를 사람이 이해할 수 있는 방식으로 제시하는 기술
- CNN = 블랙박스 → CNN 처리 과정 시각화 중요



- 각 중간 계층 ~ 최종 분류까지 입력된 이미지에서 특성이 어떻게 추출되고 학습하는지 시각적으로 설명 → 결과에 대한 신뢰 확보 가능

## 5.4.1 특성 맵(Feature map) 시각화

- **특성 맵(feature map)** : 입력 이미지 or 필터를 입력에 적용한 결과
- '특성 맵을 시각화한다' : 특성 맵에서 입력 특성을 감지하는 방법을 이해할 수 있도록 돕는 것

## 5.5 그래프 합성곱 네트워크(Graph convolutional network)

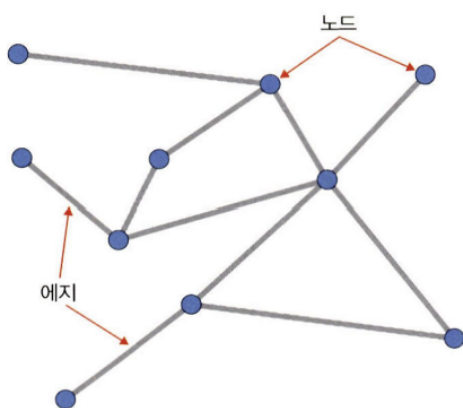
- 그래프 데이터를 위한 신경망

### 5.5.1 그래프란

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

\*node & edge : 풀고자 하는 문제에 대한 전문가 지식이나 직관 등으로 구성

- 그림과 같은 형태



#### [그래프의 구성 요소]

- **노드(node, vertex)** : 그림에서 파란색 원 - 원소

- **에지(edge)** : 두 노드를 연결한 선 - 결합 방법(single, double, triple, aromatic)

## 5.5.2 그래프 신경망(Graph Neural Network, GNN)

- 그래프 구조에서 사용하는 신경망

### [그래프 데이터 표현]

#### 1단계 ) 인접 행렬(adjacency matrix)

- 노드  $n$ 개를  $n \times n$  행렬로 표현
- 생성된 인접 행렬 내의 값은 'i와 j의 관련성 여부'를 만족하는 값으로 채움
- 컴퓨터가 이해하기 쉽게 그래프로 표현하는 과정

#### 2단계 ) 특성 행렬(feature matrix)

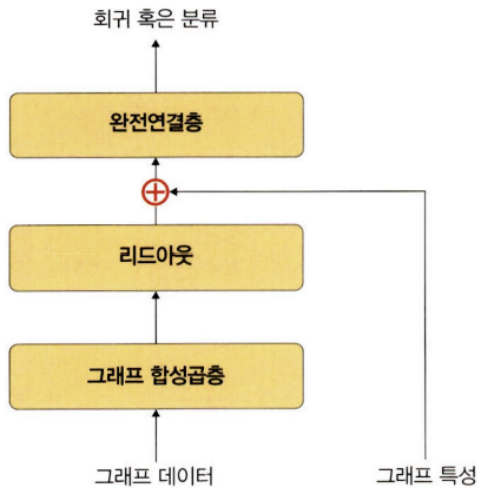
- 인접 행렬만으로 특성 파악이 어려움 → 단위 행렬 적용
- 각 입력 데이터에서 이용할 특성을 선택
- 각 행은 선택된 특성에 대해 각 노드가 갖는 값을 의미

→ 그래프 특성 추출!



## 5.5.3 그래프 합성곱 네트워크(Graph Convolutional Network, GCN)

- 이미지에 대한 합성곱을 그래프 데이터로 확장한 알고리즘



- 리드아웃(readout) : 특성 행렬을 하나의 벡터로 변환하는 함수
- 전체 노드의 특성 벡터에 대해 평균 구하고 → 그래프 전체를 표현하는 하나의 벡터 생성

### ★ 그래프 합성곱층(graph convolutional layer) ★

- 그래프 형태의 데이터를 행렬 형태의 데이터로 변환 → 딥러닝 알고리즘 적용 가능

### [GCN 사용 분야]

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