

# **MACHINE** 기계 학습 **LEARNING**

오일석 지음

## 7장. 준지도 학습과 전이 학습

# PREVIEW

## ■ 지금까지는 다음 두 조건을 만족하는 상황의 학습

- 레이블링: 훈련집합의 샘플이 모두 레이블링 되어 있거나 모두 레이블링 되어 있지 않다. 전자를 지도 학습, 후자를 비지도 학습이라고 한다.
- 과업과 도메인: 기계 학습은 하나의 과업만 해결하면 되고, 훈련집합의 샘플은 모두 같은 도메인에서 수집된다.

## ■ 현대에는 레이블이 있는 샘플과 없는 샘플이 섞인 상황이 많음

- 비용 문제로 일부만 레이블링한 탭
- 레이블 없는 샘플을 버릴 것인가, 아니면 적절히 이용하는 알고리즘을 고안할 것인가?

→ **준지도 학습**의 주제

## ■ 도메인이 다른 상황

- 예, 한국인 필체에 학습된 필기 숫자 인식기를 미국에 도입하면, 7이나 4와 같은 샘플에서 오분류 가능성 → 서로 다른 도메인
- 많은 양을 새로 수집하여 처음부터 다시 학습할 것인가, 아니면 조금만 수집하고 이전 분류기를 미세 조정할 것인가?

→ **전이 학습**의 주제

# PREVIEW

## ■ 준지도 학습과 전이 학습

- 현대 기계 학습에서 중요한 연구 주제
- 불완전한(레이블 정보가 없는) 데이터가 지닌 원천적 성질을 잘 이용해야 함
- 표현 학습은 이런 성질을 자동으로 알아내려는 시도이며, 준지도 학습과 전이 학습의 토대가 됨

## 각 절에서 다루는 내용

7.1절 매니폴드와 변화 인자를 통해 표현 학습의 중요성을 설명한다.

7.2절 딥러닝의 내부 동작과 표현을 가시화하는 여러 가지 기법을 소개한다.

7.3절 레이블링이 안 된 샘플이 다수일 때 활용하는 준지도 학습을 설명한다.

7.4절 다른 과업 또는 다른 도메인에서 제작된 분류기를 활용하는 전이 학습을 설명한다.

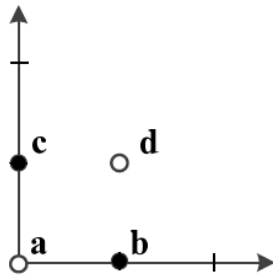
## 7.1 표현 학습의 중요성

- 7.1.1 표현 학습의 대두
  - 7.1.2 매니폴드 관찰
  - 7.1.3 프라이어를 이용한 변화 인자 풀어내기
- 
- 표현의 중요성
    - 692,688은 금액을 말할 때 편리,  $2*2*2*2*3*14431(=692,688)$ 은 공약수 구할 때 유리
    - 음력은 달의 움직임을 알아내는 데 유리하지만 1년의 실제 길이와 차이
    - 기계 학습에서도 표현이 매우 중요

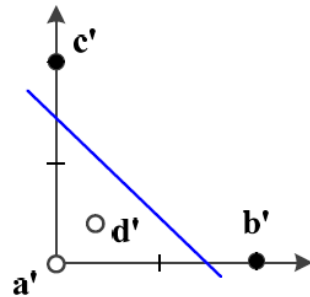
## 7.1.1 표현 학습의 대두

### ■ 기계 학습에서 좋은 표현의 중요성

- 예, [그림 1-7]과 [그림 3-9] 다시 보기

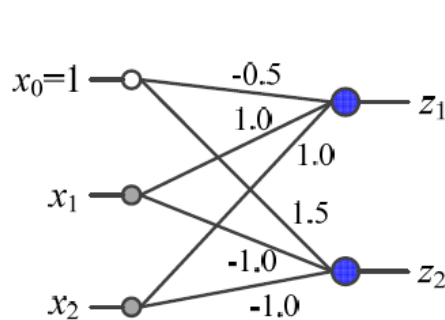


(a) 원래 특징 공간

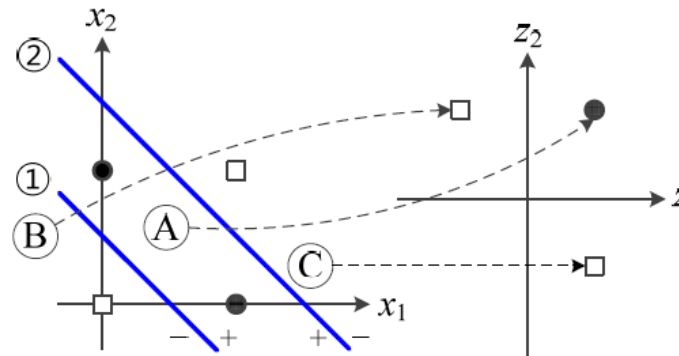


(b) 분류에 더 유리하도록 변환된 새로운 특징 공간

그림 1-7 특징 공간 변환



(a) 두 퍼셉트론을 병렬로 결합

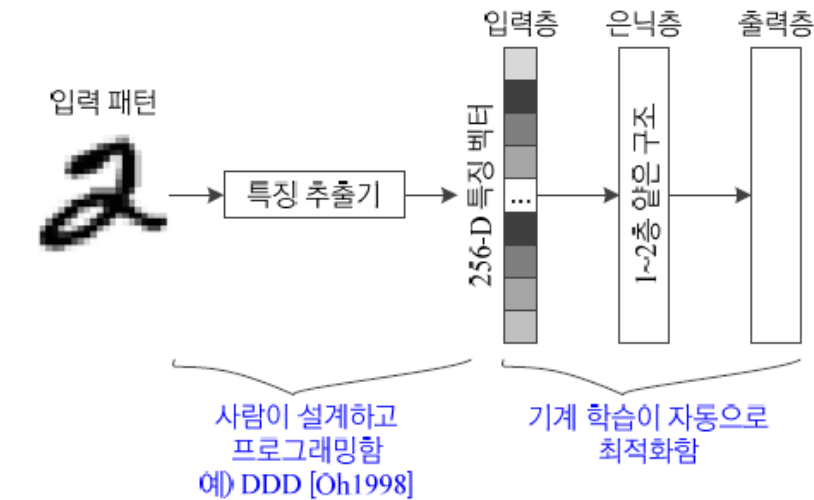


(b) 원래 특징 공간  $x$ 를 새로운 특징 공간  $z$ 로 변환

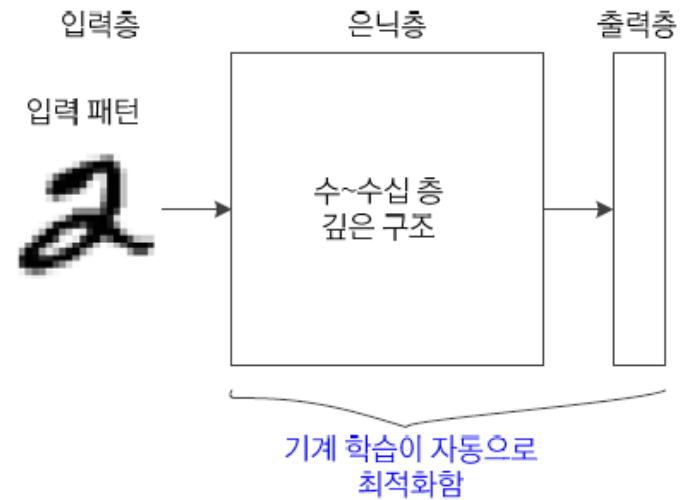
그림 3-9 특징 공간의 변환

## 7.1.1 표현 학습의 대두

- 고전적인 기계 학습(수작업 특징)과 현대적인 기계 학습(표현 학습)



(a) 고전적인 기계 학습



(b) 현대 기계 학습(딥러닝)

그림 7-1 표현 학습의 동기

- 표현 학습을 위한 문헌

- ICLR(International Conference on Learning Representations)
- 튜토리얼 논문 [Bengio2013] "Representation learning: a review and new perspective", IEEE PAMI.

## 7.1.2 매니폴드 관찰

### ■ [그림 7-2]는 $d$ 차원 특징 공간의 개념도

- 광활한 특징 공간에서 실제로 데이터가 생성되는 부분은 매우 좁음
- 가상의 매니폴드 2개를 예시하는데,  $p_1, p_2, q_1, q_2$  축은 변화 인자에 해당(예, 획의 기울음, 두께, 굵김, 잡음 정도 등)

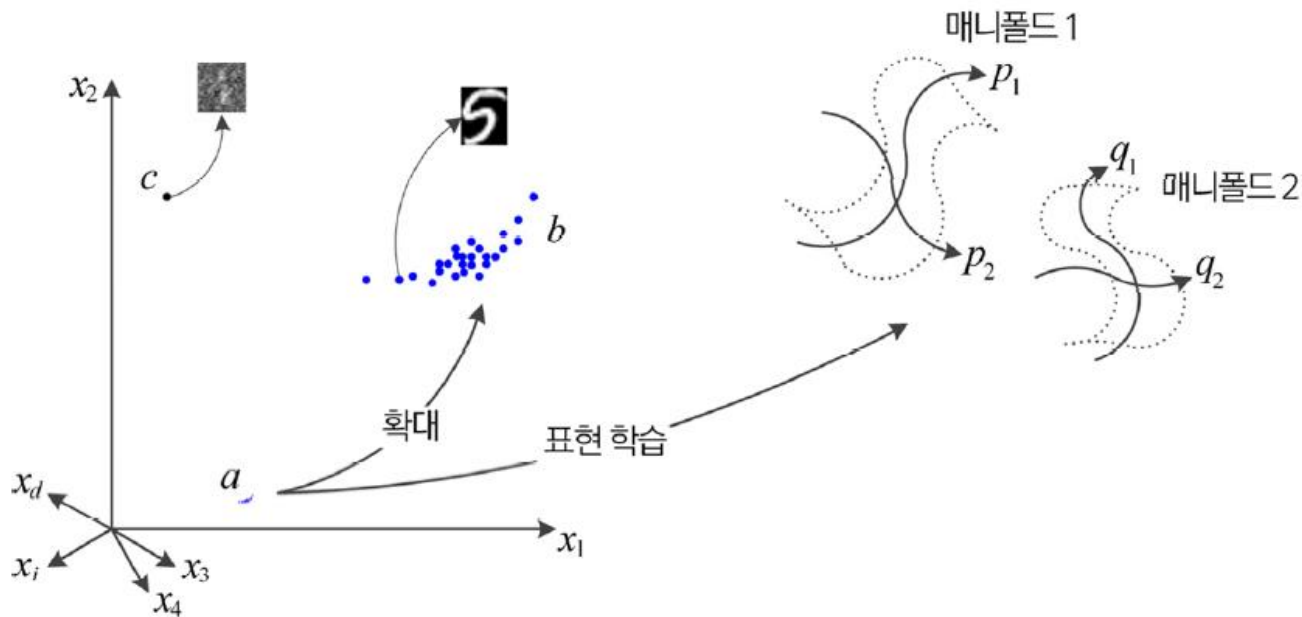


그림 7-2 데이터 생성 원천과 표현 학습



## 7.2 내부 표현의 이해

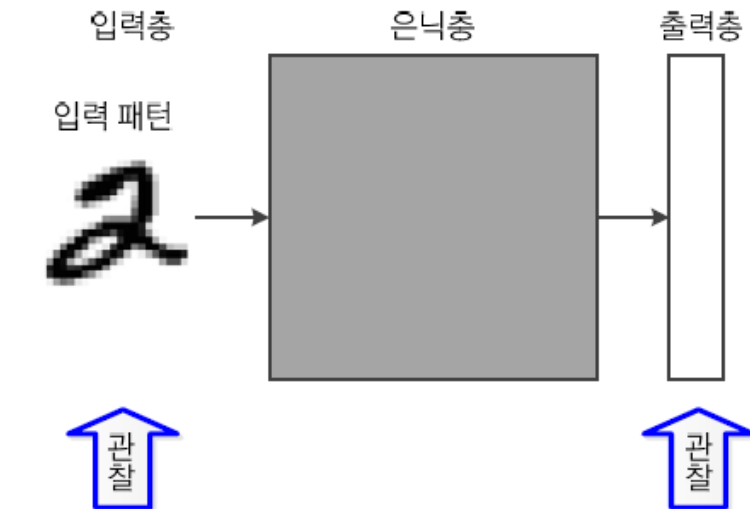
---

- 7.2.1 컨볼루션 필터의 가시화
- 7.2.2 특징 맵의 가시화
- 7.2.3 영상공간으로 역투영

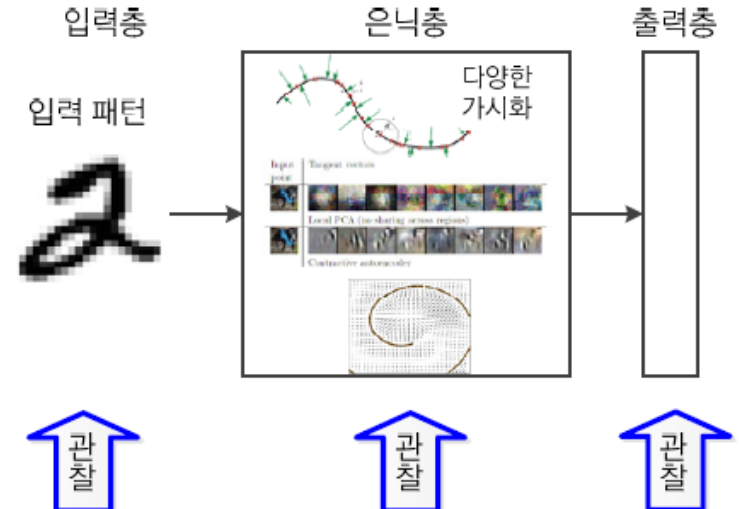
## 7.2 내부 표현의 이해

### ■ 표현의 가시화(현대 기계 학습의 중요한 추세)

- 블랙박스로 간주되던 신경망의 내부를 가시화하여, 성능에 대한 통찰력을 얻고 구조나 하이퍼 매개변수를 최적화하는 데 활용
- 준지도 학습 또는 전이 학습을 설계하는 데 길잡이



(a) 예전에는 내부 표현을 블랙박스로 간주함



(b) 현대 기계 학습은 내부 표현을 관찰하고 분석함

그림 7-4 현대 기계 학습에서 표현의 가시화

## 7.2.1 컨볼루션 필터의 가시화

### ■ 신경망 내부 표현을 가시화 하는 여러 가지 방법

- 필터 가시화
- 특징 맵 가시화
- 역투영 가시화

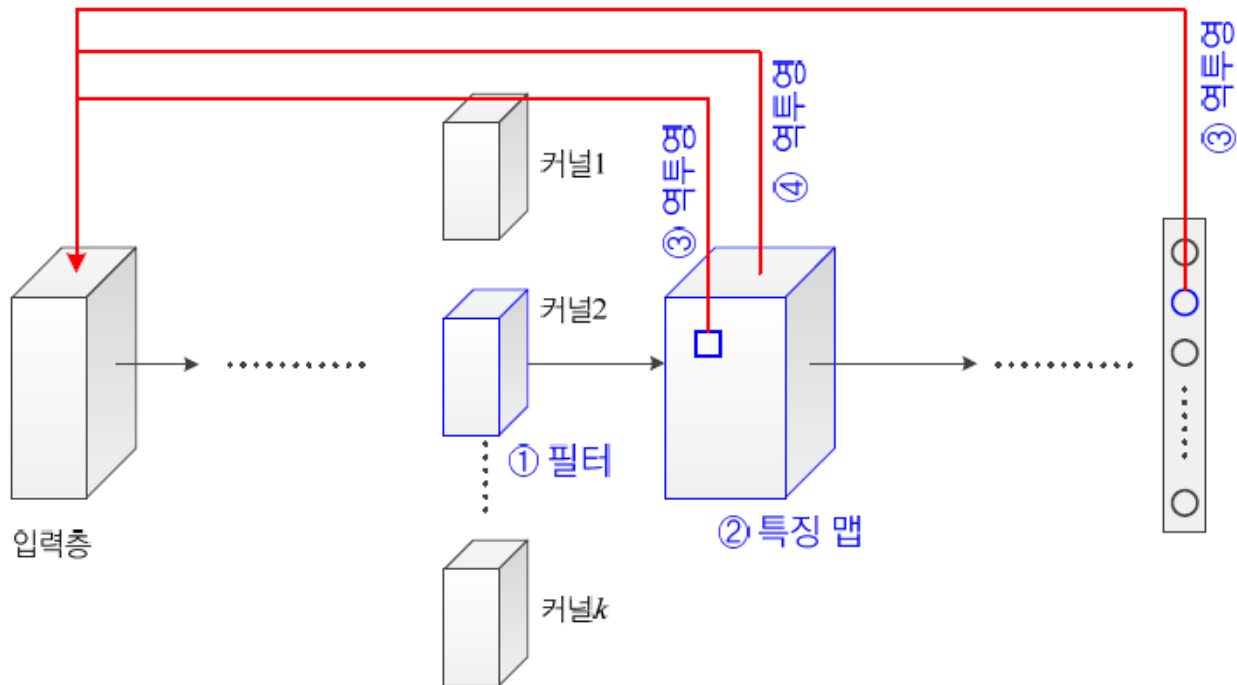


그림 7-5 CNN 가시화의 네 가지 형태

## 7.2.1 컨볼루션 필터의 가시화

### ■ 필터의 가시화 예([그림 7-6])

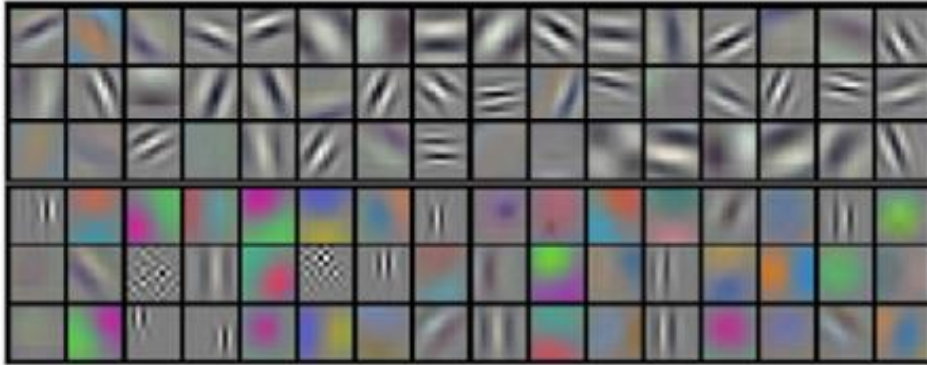


그림 7-6 첫 번째 컨볼루션 층 필터

### ■ 관찰 결과

- 첫 번째 컨볼루션 층에서는 에지나 블롭이 주로 나타남
- 영상의 종류에 무관하게 나타나는 일반적인 현상으로 밝혀짐

## 7.2.2 특징 맵의 가시화

### ■ 가시화 도구 예([그림 7-7])

- 층과 특징 맵을 마우스로 쉽게 선택할 수 있음
- 선택된 녹색 맵에서는 고양이 얼굴이 활성화되었음을 알 수 있음

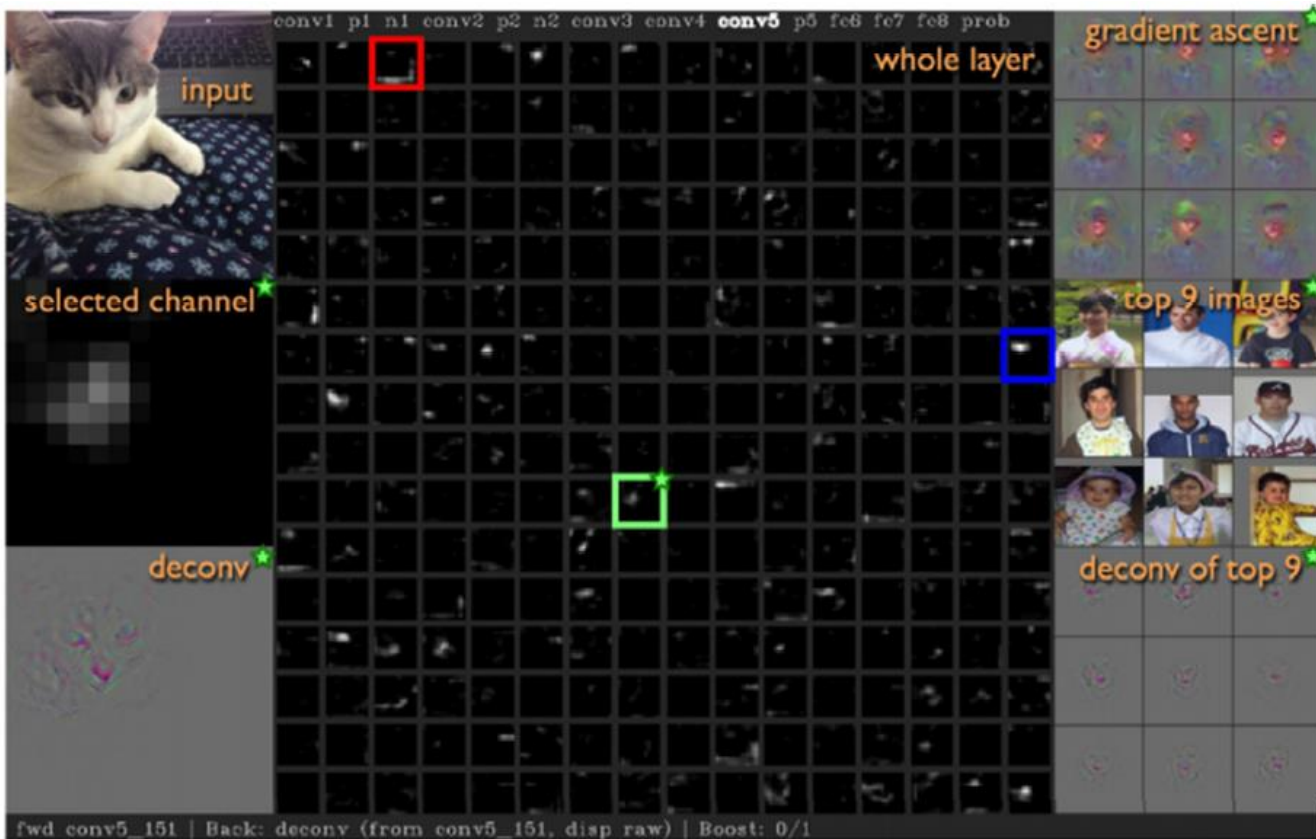


그림 7-7 가시화 도구 화면

## 7.2.2 특징 맵의 가시화

- 다른 영상을 입력해도 녹색 특징 맵은 얼굴 부위가 활성화됨을 확인할 수 있음

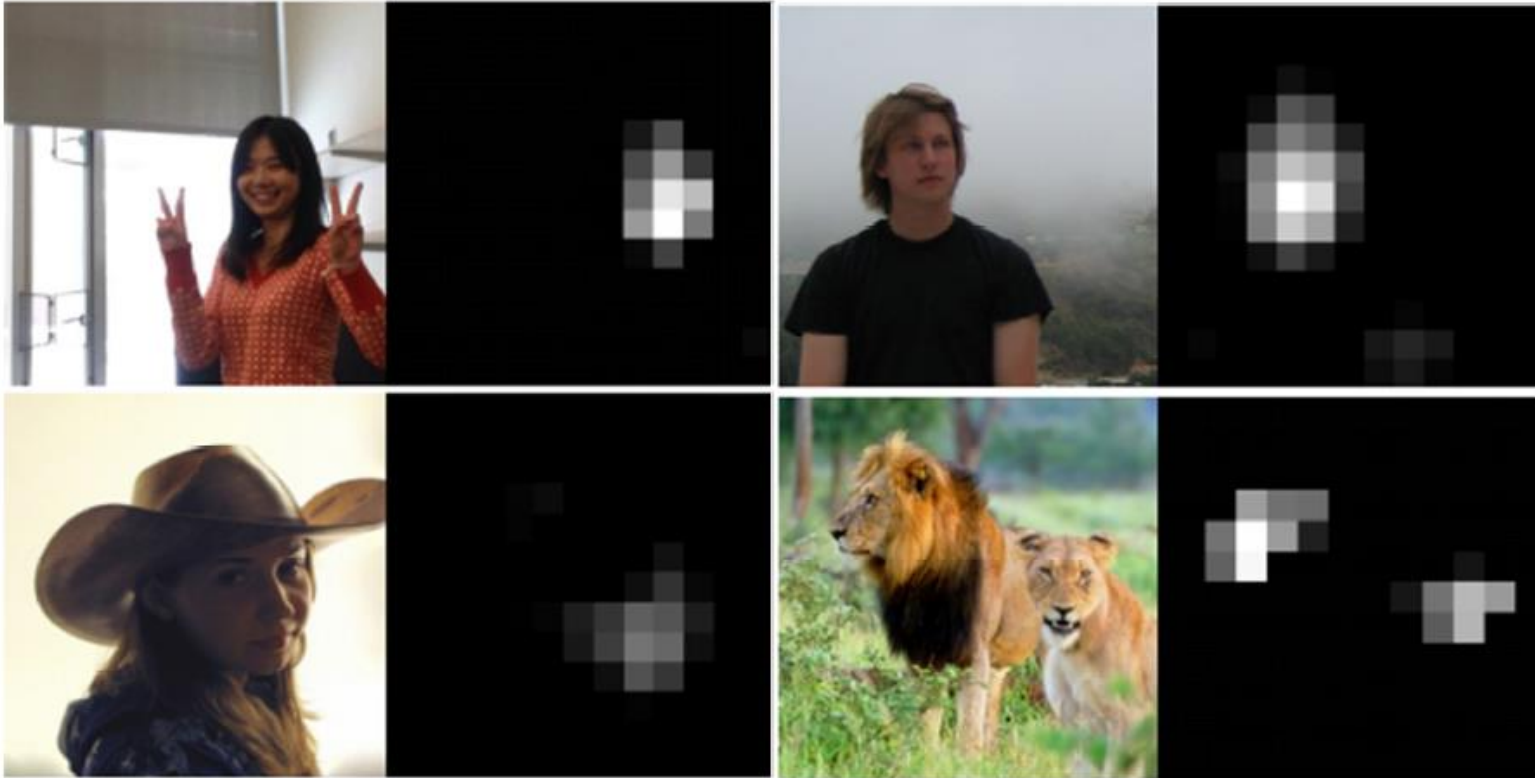


그림 7-8 얼굴을 검출하는 특징 맵

## 7.2.3 영상 공간으로 역투영

### ■ 가시화의 두 가지 방식

- 앞의 두 가지 가시화 기법은 전방 계산 과정에서 발생하는 필터 또는 특징 맵을 보여줌
- 역투영 기법은 특정 노드(뉴런)를 활성화하는 입력 공간의 신호를 알아내어 보여줌

### ■ 최적화를 이용한 역투영

- 관찰 대상 노드를  $i$ 라 하고  $a_i(\mathbf{x})$ 를 영상  $\mathbf{x}$ 가 입력되었을 때  $i$ 의 활성값이라 하면, 역투영 문제는 식 (7.1)로 표현됨

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}} a_i(\mathbf{x}) \quad (7.1)$$

- 식 (7.1)의 최적화 문제를 식 (7.2)의 경사 상승법으로 풀( $\mathbf{x}_0$ 는 난수 영상)

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \eta \frac{\partial a_i(\mathbf{x})}{\partial \mathbf{x}} \quad (7.2)$$

- 실제로는 여러 가지 규제 기법을 적용하여 풀( $r_\theta$ 는 규제 함수)

$$\mathbf{x}_{t+1} = r_\theta \left( \mathbf{x}_t + \eta \frac{\partial a_i(\mathbf{x})}{\partial \mathbf{x}} \right) \quad (7.5)$$

- [그림 7-7]에서 "gradient ascent"라 쓰인 영역이 이렇게 찾은 영상을 보여줌

## 7.2.3 영상 공간으로 역투영

### ■ 디컨볼루션을 이용한 역투영

- 입력 영상  $I$ 를 주고 관찰 대상 노드를  $i$ 를 지정하면,  $i$ 가 속한 층에서 출발하여 디컨볼루션(컨볼루션의 역 연산을 적용)을 수행하여  $I^{-1}$ 를 보여줌
- [그림 7-7]에서 "deconv"와 "deconv of top 9"이라 쓰인 영역이 이렇게 찾은 영상을 보여줌

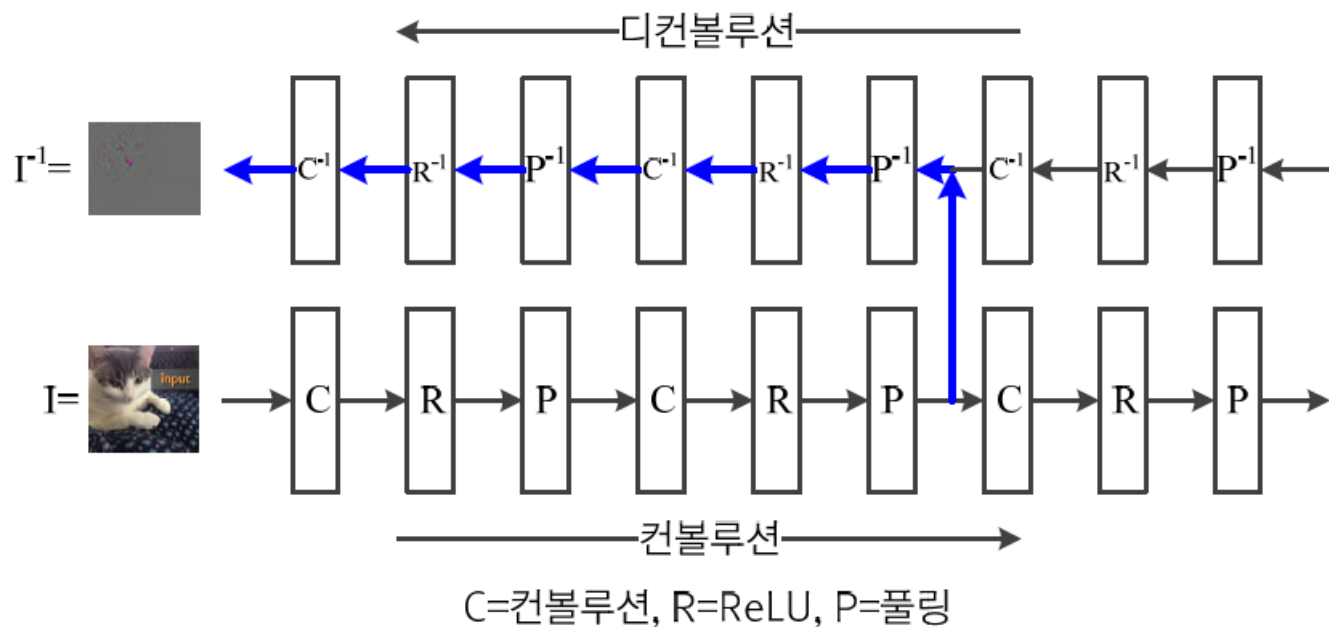


그림 7-9 디컨볼루션의 동작 원리



## 7.3 준지도 학습

### ■ 7.3.1 동기과 원리

### ■ 7.3.2 알고리즘

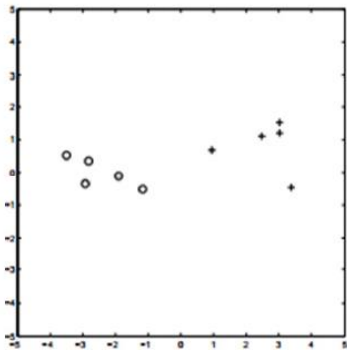
### ■ 준지도 학습이 사용하는 훈련집합

- 레이블이 있는  $\mathbb{X}_l = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ ,  $\mathbb{Y}_l = \{y_1, y_2, \dots, y_n\}$ 과 레이블이 없는  $\mathbb{X}_u = \{\mathbf{x}_{n+1}, \mathbf{x}_{n+2}, \dots, \mathbf{x}_{n+m}\}$
- 보통  $m \gg n$ , 즉 레이블이 없는 샘플이 훨씬 많음

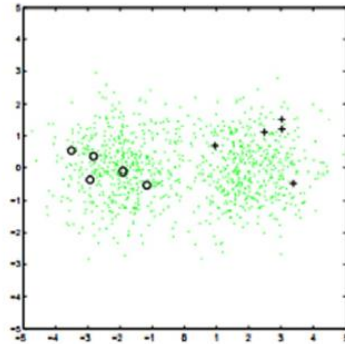
## 7.3.1 동기와 원리

### ■ 레이블이 없는 데이터가 정말 도움이 되는가

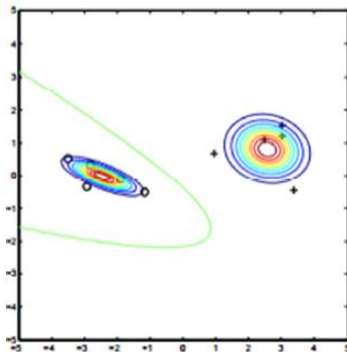
- 그릴 수도([그림 7-10]) 아닐 수도 있음([그림 7-11])
- '주어진 데이터의 구조에 적합한 모델을 사용하는' 경우 성능 향상



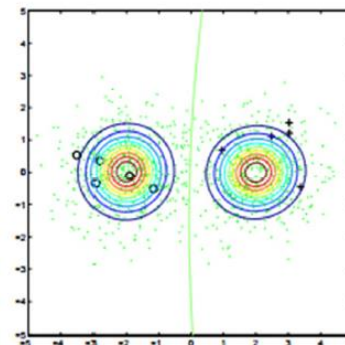
(a) 레이블링된 데이터



(b) 레이블링된 데이터와 레이블링이 안 된 데이터



(c) 지도 학습으로 추정된 모델



(d) 준지도 학습으로 추정된 모델

그림 7-10 레이블이 없는 데이터  $x_u$ 가 도움이 되는 상황

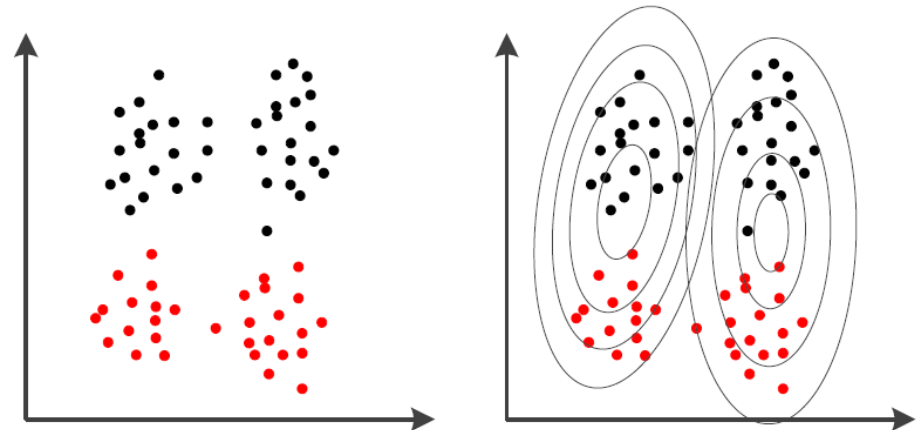


그림 7-11 문제 구조와 모델이 잘 맞지 않아 성능이 저하된 상황

## 7.3.2 알고리즘

### ■ 생성 모델

1.  $\mathbb{X}_l$ 과  $\mathbb{X}_u$ 를 가지고  $P(\mathbf{x})$ 를 추정 ( $\mathbb{Y}_l$ 을 무시한 비지도 학습). 이때 6.4.2절의 가우시안 혼합을 사용한다.
2. 각각의 가우시안은 자신에 속한 샘플을 보고 부류를 정한다.
3. 각 부류는 자신에 속한 가우시안으로  $P(\mathbf{x}|y)$ 를 추정한다.

### ■ 적용에 한계

- 가우시안 분포에 맞는 데이터로 국한해야 함
- 가우시안 혼합을 적용할 때 가우시안 개수를 정확하게 알기 어려움

## 7.3.2 알고리즘

### ■ 현대적 생성 모델

- 4.5.2절에서 소개한 생성 모델 GAN을 사용
- 가짜 샘플에 해당하는  $c+1$ 이라는 레이블을 추가로 사용
- 분별기 D의 목적함수는 세 가지 항을 가짐
  - 가짜 샘플을  $c+1$ 에 배정하는 항
  - $\mathbb{X}_u$  샘플이  $c+1$ 에 배정되는 것을 막는 항
  - $\mathbb{X}_l$  샘플을 해당 부류로 배정하는 항

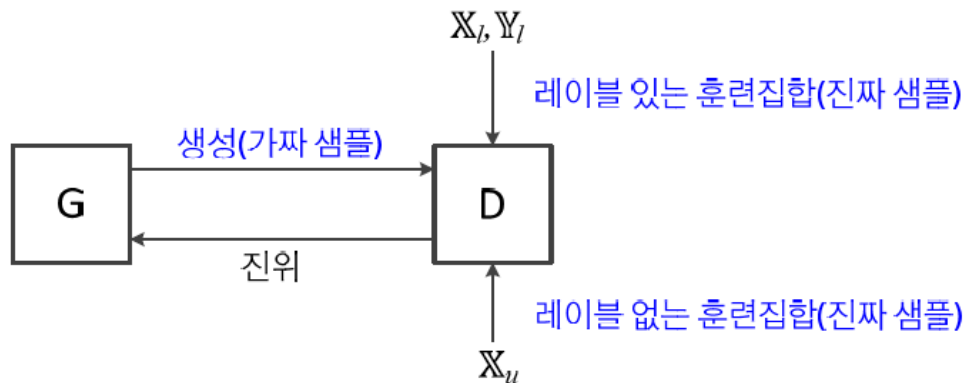


그림 7-12 GAN을 이용한 준지도 학습

## 7.3.2 알고리즘

### ■ 자가 학습 self learning

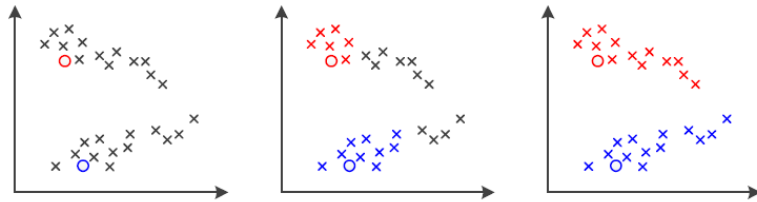
#### 알고리즘 7-1 자가 학습

입력:  $\mathbb{X}_l = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ ,  $\mathbb{Y}_l = \{y_1, y_2, \dots, y_n\}$ ,  $\mathbb{X}_u = \{\mathbf{x}_{n+1}, \mathbf{x}_{n+2}, \dots, \mathbf{x}_{n+m}\}$ , 학습기  $f$ , 선택 비율  $r$

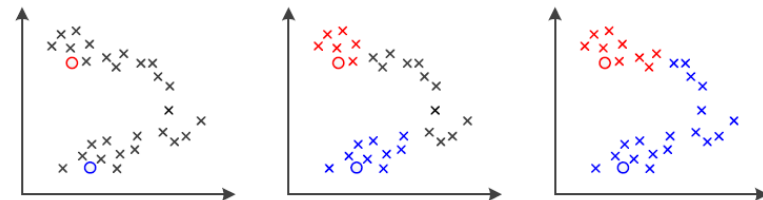
출력:  $\mathbb{Y}_u = \{y_{n+1}, y_{n+2}, \dots, y_{n+m}\}$ , 학습된  $f$

- 1 while (not 멈춤 조건)
- 2     $\mathbb{X}_l$ 과  $\mathbb{Y}_l$ 을 훈련집합으로 사용하여 학습기  $f$ 를 학습한다.
- 3     $f$ 로  $\mathbb{X}_u$ 를 분류하여  $\mathbb{Y}_u$ 를 알아낸다.
- 4    분류된  $\mathbb{X}_u$  중 신뢰도가 높은 순으로  $r\%$ 의 부분집합  $\mathbb{X}_{u\_subset}$ 을 선발한다.
- 5     $\mathbb{X}_u = \mathbb{X}_u - \mathbb{X}_{u\_subset}$ ,  $\mathbb{X}_l = \mathbb{X}_l \cup \mathbb{X}_{u\_subset}$ ,  $\mathbb{Y}_l = \mathbb{Y}_l \cup \mathbb{Y}_{u\_subset}$

### ■ 소속이 애매한 샘플에 민감한 상황 발생



(a) 잘 작동하는 상황



(b) 소속이 애매한 샘플에 민감한 상황

○ 표시된 점은 레이블링 된 샘플  
x 표시된 점은 레이블링 안 된 샘플

그림 7-13 자가 학습의 예

## 7.3.2 알고리즘

### ■ 협동 학습 co-training

- 학습기 2개가 서로 협동하여  $\mathbb{X}_{expand}$ 를 확장하면서 발전해 감

#### 알고리즘 7-2 협동 학습(2시점 협동 학습)

입력:  $\mathbb{X}_l = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ ,  $\mathbb{Y}_l = \{y_1, y_2, \dots, y_n\}$ ,  $\mathbb{X}_u = \{\mathbf{x}_{n+1}, \mathbf{x}_{n+2}, \dots, \mathbf{x}_{n+m}\}$ , 학습기  $f$ , 선발 개수  $q$

출력:  $\mathbb{Y}_u = \{y_{n+1}, y_{n+2}, \dots, y_{n+m}\}$ , 학습된  $f$

- 1 특징 공간  $\mathbf{x}$ 를 2개의 부분공간  $\mathbf{x}_{subset1}$ 과  $\mathbf{x}_{subset2}$ 로 구분한다.
- 2  $\mathbb{X}_{expand} = \mathbb{X}_l$
- 3 while ( $\mathbb{X}_u \neq \emptyset$ )
- 4      $\mathbb{X}_{expand}$ 의  $\mathbf{x}_{subset1}$  시점으로 학습기  $f_1$ 을 학습한다.
- 5      $\mathbb{X}_{expand}$ 의  $\mathbf{x}_{subset2}$  시점으로 학습기  $f_2$ 를 학습한다.
- 6      $f_1$ 로  $\mathbb{X}_u$ 를 분류한다.
- 7      $f_2$ 로  $\mathbb{X}_u$ 를 분류한다.
- 8      $f_1$ 의 분류 결과에서 신뢰도가 가장 높은  $q$ 개 샘플을  $\mathbb{X}_u$ 에서  $\mathbb{X}_{expand}$ 로 옮긴다.
- 9      $f_2$ 의 분류 결과에서 신뢰도가 가장 높은  $q$ 개 샘플을  $\mathbb{X}_u$ 에서  $\mathbb{X}_{expand}$ 로 옮긴다.
- 10    8~9에서 옮긴 샘플의 레이블을  $\mathbb{Y}_u$ 에 추가한다.

## 7.3.2 알고리즘

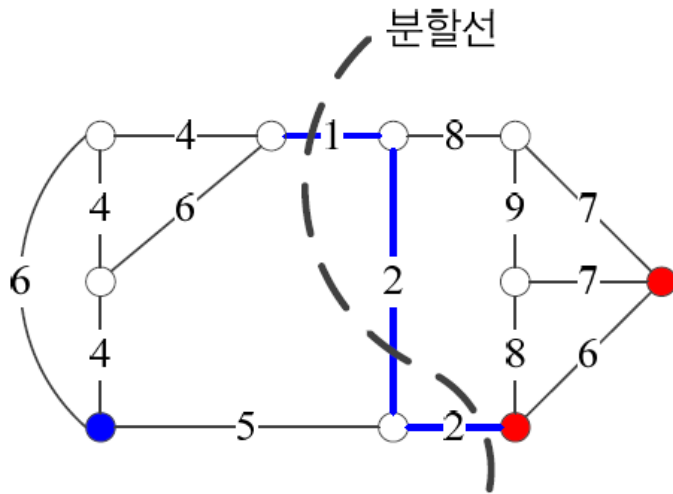
- 학습기 2개가 서로 가르치는 방식으로 수정하면

2.  $\mathbb{X}_{expand1} = \mathbb{X}_l, \mathbb{X}_{expand2} = \mathbb{X}_l$
3. while ( $\mathbb{X}_u \neq \emptyset$ )
4.  $\mathbb{X}_{expand1}$ 의  $\mathbf{x}_{subset1}$  시점으로 학습기  $f_1$ 을 학습한다.
5.  $\mathbb{X}_{expand2}$ 의  $\mathbf{x}_{subset2}$  시점으로 학습기  $f_2$ 를 학습한다.
6.  $f_1$ 로  $\mathbb{X}_u$ 를 분류한다.
7.  $f_2$ 로  $\mathbb{X}_u$ 를 분류한다.
8.  $f_1$ 의 분류 결과에서 신뢰도가 가장 높은  $q$ 개 샘플을  $\mathbb{X}_u$ 에서  $\mathbb{X}_{expand2}$ 로 옮긴다.
9.  $f_2$ 의 분류 결과에서 신뢰도가 가장 높은  $q$ 개 샘플을  $\mathbb{X}_u$ 에서  $\mathbb{X}_{expand1}$ 로 옮긴다.

## 7.3.2 알고리즘

### ■ 그래프 방법

- 샘플 사이의 유사도에 따라 그래프를 구성
  - 예, 샘플마다  $k$ 개의 최근접 이웃을 찾아 에지로 이어줌
  - 복잡한 비선형 분포를 반영하기 위해 정교한 그래프 구축 방법 필요
- 최소 분할 적용하여 분할선 찾음
- 같은 부분집합에 속하는 샘플에 같은 부류 레이블 부여



빨간색과 파란색 노드는 레이블링 된 샘플  
흰색 노드는 레이블링 안 된 샘플

그림 7-14 그래프 최소 분할



## 7.3.2 알고리즘

### ■ 표현 변환

단계 1.  $\mathbb{X}_l$ 과  $\mathbb{X}_u$ 를 합친 데이터로 주어진 목적을 달성하는 데 더 유리한 표현으로 변환한다.

단계 2.  $\mathbb{X}_l$ 을 새로운 표현으로 변환한다.

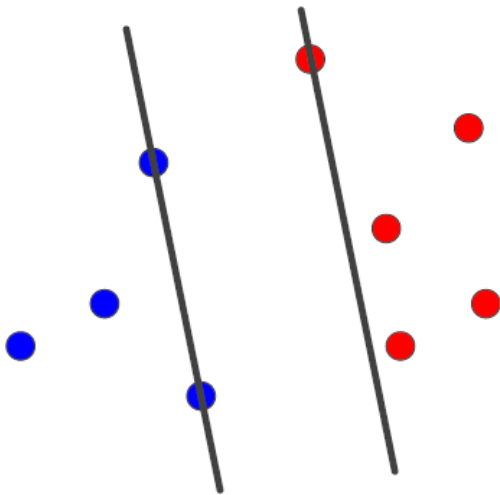
단계 3. 변환된  $\mathbb{X}_l$ 과  $\mathbb{Y}_l$ 로 목적에 따라 기계 학습을 수행한다.

- 단계 1을 구현하는 방법
  - 6.5~6.6절 기법 → 얇은 변환이라는 한계
  - 6.7.2절의 적층 오토인코더 → 깊은 변환

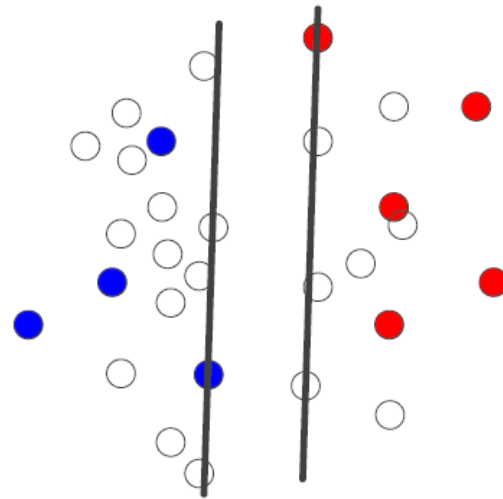
## 7.3.2 알고리즘

### ■ 밀집 지역 회피

- 결정 경계가 밀집 지역을 지나면 오분류 가능성 높아짐 → 밀집 지역을 회피하여 결정 경계를 정함
- 예, TSVM(트랜스덕티브 SVM)



(a)  $\mathbb{X}_l, \mathbb{Y}_l$ 로 지도 학습을 수행한 SVM



(b)  $\mathbb{X}_l$ 과  $\mathbb{X}_u$ 를 모두 이용한 준지도 학습으로 구한 TSVM

그림 7-15 밀집지역 회피를 이용한 TSVM

## 7.4 전이 학습

### ■ 7.4.1 과업 전이

### ■ 7.4.2 도메인 전이

### ■ 일상 생활에서 전이 학습

- 피아노를 칠 줄 아는 사람은 못 치는 사람보다 바이올린을 빨리 배움
- C언어에 익숙한 학생은 파이썬을 금방 배움
- 두 영역의 공통 지식을 공유하기 때문

### ■ 기계 학습에서 전이 학습

- 어떤 도메인에서 제작한 프로그램을 데이터가 적어 애를 먹는 새로운 도메인에 적용하여 높은 성능을 얻는 기법
- 현대 기계 학습에서 널리 활용되고 있음

## 7.4 전이 학습

### ■ 과업이 다른 경우와 도메인이 다른 경우로 구분

- 과업<sup>task</sup>이 달라지는 경우: 영상 인식에서 음성 인식으로 전이하는 것처럼 응용분야가 달라질 수 있다. 이는 두 과업 사이의 거리가 아주 먼 상황이며, 아직 이러한 상황에 적용할 수 있는 실용적인 연구 결과는 없다. 반면, 자연영상을 1,000부류로 인식하는 과업을 200종의 나뭇잎을 인식하는 과업으로 전이하는 경우를 생각할 수 있다. 응용분야가 다른 경우보다는 과업 간 거리가 훨씬 짧다. 현재 이런 종류의 전이 학습은 보편적으로 적용할 수 있는 기술이 되었다. 예를 들어, 공개된 VGGNet을 소량의 나뭇잎 영상만 있는 상황에 전이하여 실용적인 나뭇잎 인식 프로그램을 만들어 앱 스토어에 공개한 사례가 여럿 있다.<sup>5</sup>
- 도메인<sup>domain</sup>이 달라지는 경우: 특징 공간이 다른 경우와 특징 공간은 같은데 데이터의 확률분포가 다른 경우로 구분한다. 전자 사례로는 영불 번역기를 영한 번역기로 전이하거나 한국어 정보 검색기를 베트남어 정보 검색기로 전이하는 상황을 들 수 있다. 두 도메인은 단어 집합이 달라 특징 공간이 다를 수 밖에 없다. 후자 사례는 한국인이 쓴 필기 숫자 데이터베이스로 만든 인식기를 인도에 수출하는 상황에서 발생할 수 있다. 두 도메인은 같은 크기의 숫자 영상을 사용하면 되는데, 필기 습관이 달라 데이터의 확률분포가 다를 것이다. 이때 전이 학습을 사용한다면, 인도인을 대상으로 소량의 데이터만 수집해도 높은 성능을 얻을 수 있다.

## 7.4.1 과업 전이

### ■ 과업 전이의 성공 사례

- 2012년 ILSVRC대회에서 AlexNet은 오류율 16.3%의 경이로운 성능 달성
  - 딥러닝의 가능성을 입증
  - 프로그램, 가중치, 하이퍼 매개변수 등 **모든 것을 공개**하여 전이 학습을 개척함
- 영상 인식에서 과업 전이가 성공함으로써 전이 학습이 뜨거운 연구 주제로 부상

## 7.4.1 과업 전이

### ■ 기성 CNN 특징

- 성공적으로 학습된 신경망의 특징 추출 부분을 다른 과업에 활용

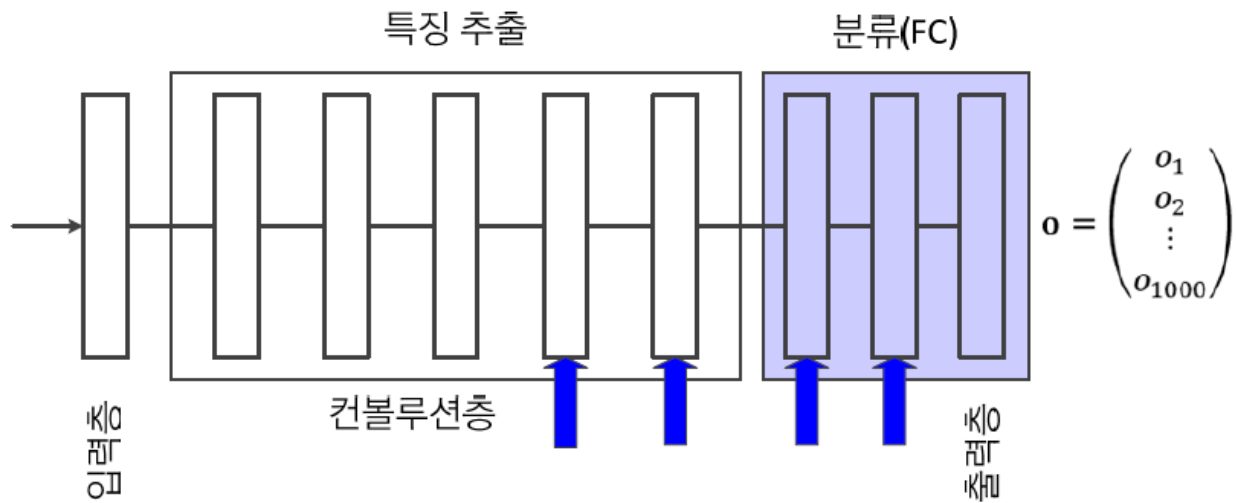


그림 7-16 AlexNet에서의 전이 학습(파란색으로 표시된 분류층 3개를 새로운 과업에 맞게 대체함)

## 7.4.1 과업 전이

### ■ 동결 방식

- [그림 7-16]의 파란색 실선 화살표로 표시된 층 중 하나를 골라 특징을 취함
- 이 특징은 컨볼루션 층을 여럿 통과하면서 정제되었으므로 얇은 신경망(예, MLP)을 사용해도 높은 성능으로 분류할 수 있음

#### 동결 방식:

1. 새로운 과업의 훈련집합  $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ 의 샘플을 하나씩 [그림 7-16]의 AlexNet에 입력하여 새로운 훈련집합  $\mathbb{X}' = \{\mathbf{x}'_1, \mathbf{x}'_2, \dots, \mathbf{x}'_n\}$ 을 만든다.
2.  $\mathbb{X}' = \{\mathbf{x}'_1, \mathbf{x}'_2, \dots, \mathbf{x}'_n\}$ 과  $\mathbb{Y} = \{y_1, y_2, \dots, y_n\}$ 을 훈련집합으로 사용하여 학습기를 학습한다.

## 7.4.1 과업 전이

### ■ 미세 조정 방식

- [그림 7-16]의 FC 부분을 떼어낸 후, 새로운 구조를 덧붙여 다시 학습
- 이때 학습률을 낮게 설정해야 함(높으면 원래 가중치가 훼손)

#### 미세 조정 방식:

1. [그림 7-16]에서 파란색으로 표시된 곳을 떼어낸 후 새로운 구조를 덧붙인다. 이때 새로운 과업에 맞게 출력 노드의 개수를 설정한다.
2. 덧붙인 곳의 가중치를 초기화하고, 하이퍼 매개변수를 적절하게 설정한다.
3. 새로운 과업의 훈련집합  $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ 와  $\mathbb{Y} = \{y_1, y_2, \dots, y_n\}$ 을 훈련집합으로 사용하여 개조된 AlexNet을 학습한다.



## 7.4.1 과업 전이

### ■ 이후 발전한 기성 CNN

표 7-1 기성 CNN

기성 CNN	문헌	특성	개발 그룹
AlexNet	[Krizhevsky2012], [Jia2014]	2012년 ILSVRC 우승, 8층 구조	토론토대학
VGGNet	[Simonyan2015]	2014년 ILSVRC 준우승, 19층 구조	옥스퍼드대학
GoogLeNet	[Szegedy2015, Szegedy2016]	2014년 ILSVRC 우승, 27층 구조	구글
ResNet	[He2016a]	2015년 ILSVRC 우승, 152층 구조	마이크로소프트
R-CNN	[Girshick2015]	물체 검출용 CNN	마이크로소프트

### ■ 왜 작동할까?

- [Yosinski2014]의 의미 있는 설명
- 하지만 불충분한 설명 → 미래의 좋은 연구 주제

## 7.4.2 도메인 전이

### ■ 도메인 전이

- 과업은 같은데(즉 레이블 공간이 같음), 도메인이 다른 상황
- 특징 공간이 다른 경우와 특징 공간은 같은데 확률분포가 다른 경우로 나뉨
- 후자를 도메인 적응이라 부름

### ■ 도메인 적응 domain adaptation

- 예, 나뭇잎 인식
  - 원천 도메인은 따낸 나뭇잎 → 목표 도메인은 나무에 붙어있는 나뭇잎 영상



(a) 도메인 1



(b) 도메인 2

그림 7-17 도메인 적응이 필요한 예제 시나리오

## 7.4.2 도메인 전이

### ■ 도메인 전이 방법

- 원천 도메인의 훈련집합  $\mathbb{X}_s = \{\mathbf{x}_{s1}, \mathbf{x}_{s2}, \dots, \mathbf{x}_{sn}\}$ ,  $\mathbb{Y}_s$ , 목표 도메인의 훈련집합  $\mathbb{X}_t = \{\mathbf{x}_{t1}, \mathbf{x}_{t2}, \dots, \mathbf{x}_{tn}\}$
- 목표 도메인을 위한 레이블 정보  $\mathbb{Y}_t$ 는 있을 경우(지도 도메인 적응), 없는 경우(비지도 도메인 적응), 일부만 있는 경우(준지도 도메인 적응)로 나뉜다

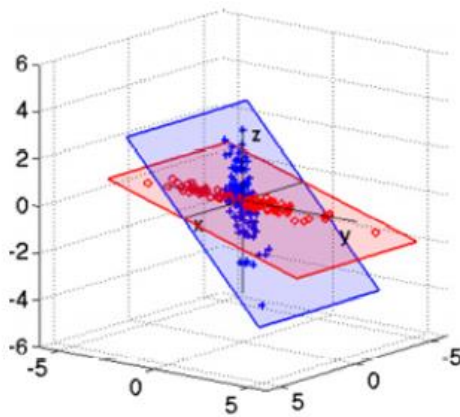
### ■ [Daume2009] 방법

- 특징 공간을 3배로 확장하여 두 도메인의 확률분포를 맞춤

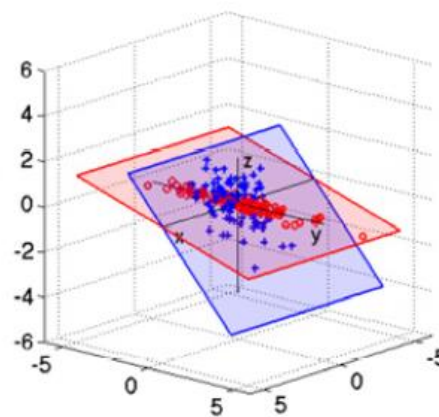
$$\left. \begin{aligned} \Phi^s(\mathbf{x}_s) &= (\mathbf{x}_s^T, \mathbf{x}_s^T, \mathbf{0})^T \\ \Phi^t(\mathbf{x}_t) &= (\mathbf{x}_t^T, \mathbf{0}, \mathbf{x}_t^T)^T \end{aligned} \right\} \quad (7.7)$$

## 7.4.2 도메인 전이

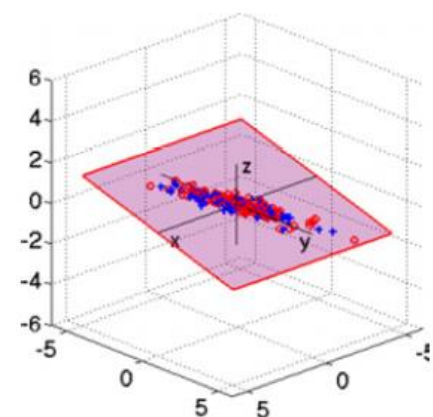
- [Sun2016] 방법([Daume2009]를 비지도 도메인 적응으로 확장)
  - [그림 7-18]에서 파란 점은 원천 도메인 샘플, 빨간 점은 목표 도메인 샘플
  - 화이트닝 변환과 컬러링 변환으로 두 도메인의 확률분포를 맞춤



(a)  $\mathbb{X}_s$ 와  $\mathbb{X}_t$



(b)  $\mathbb{X}_s$ 를 화이트닝 변환하여  $\mathbb{X}'_s$  구축



(c)  $\mathbb{X}'_s$ 를  $\mathbb{X}_t$ 로 컬러링함

그림 7-18 비지도 도메인 적응

## 7.4.2 도메인 전이

- 알고리즘 형태로 쓰면,

### 알고리즘 7-3 비지도 도메인 적응

**입력:** 원천 도메인 설계행렬  $\mathbf{X}_s$ , 목표 도메인 설계행렬  $\mathbf{X}_t$ , 규제 계수  $\lambda$

// 두 설계행렬은 원점 중심으로 분포한다고 가정함(즉, [그림 5-7]의 첫 번째 변환이 적용됨)

**출력:** 변환된 원천 도메인 설계행렬  $\mathbf{X}_s^*$

```
1   $\Sigma'_s = \Sigma_s + \lambda \mathbf{I}$ 
2   $\Sigma'_t = \Sigma_t + \lambda \mathbf{I}$ 
3   $\mathbf{X}'_s = \mathbf{X}_s (\Sigma'_s)^{-\frac{1}{2}}$            // 화이트닝
4   $\mathbf{X}_s^* = \mathbf{X}'_s (\Sigma'_t)^{\frac{1}{2}}$        // 컬러링
```

- [알고리즘 7-3]으로 변환한 훈련집합  $\mathbf{X}_s^*, \mathbf{Y}_s$ 를 가지고 학습기를 학습한 후, 목표 도메인에 활용