

$$a^0 = 1 [a^0]$$

$$\arcsin(2)$$

## 11.2. 사전훈련된 층 재사용하기 (전이학습 Transfer learning)

$$x_{n+1} =$$

# 전이 학습 Transfer learning

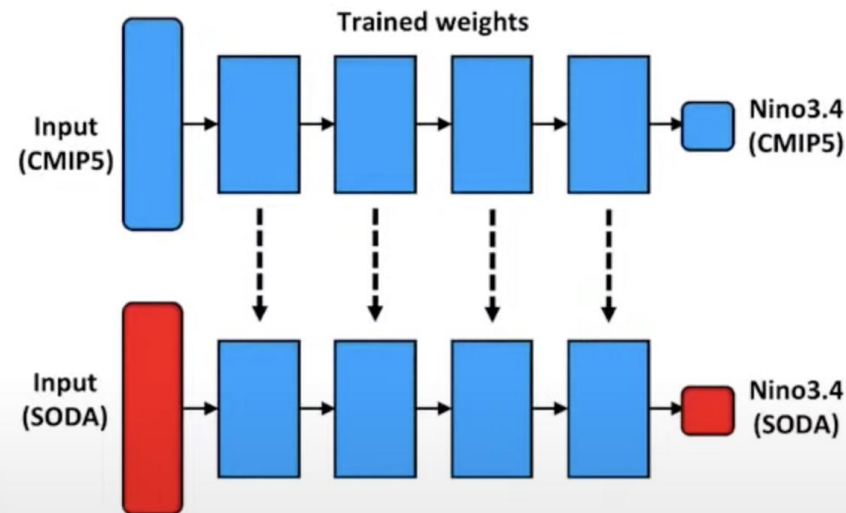
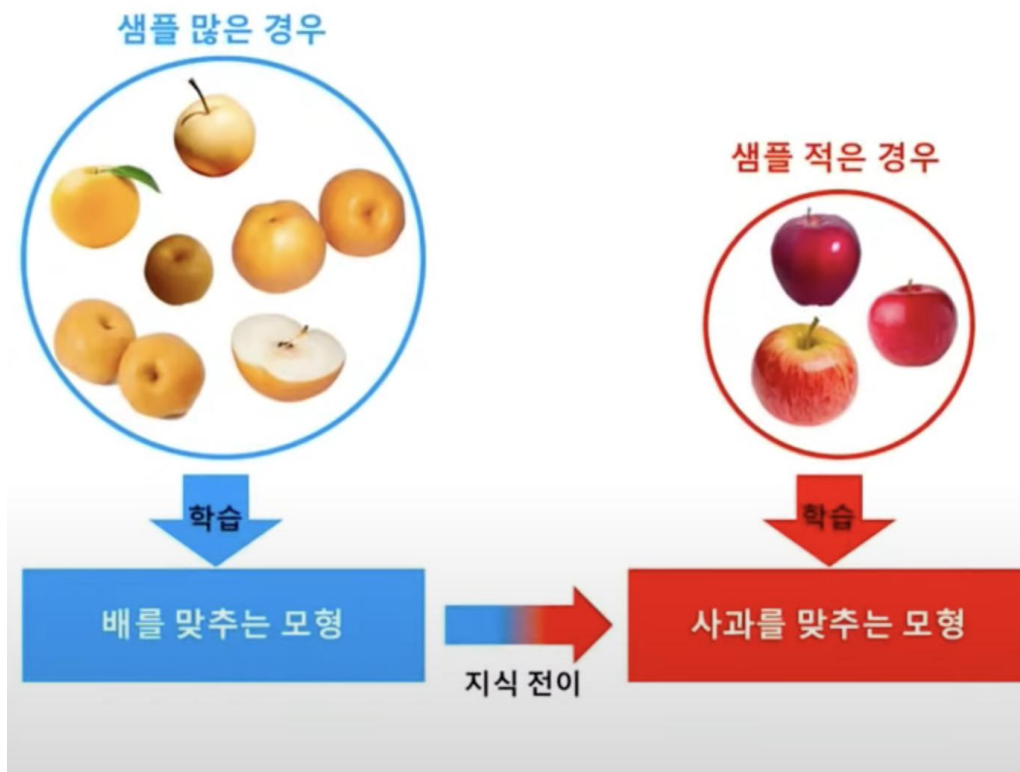
## ✓ 필요성

- 일반적으로 **큰 규모**의 심층 신경망(DNN)을 처음부터 새로 훈련하는 것은 어려움
- ➔ **비슷한 유형의 문제**를 처리하는 신경망이 존재하는지 확인 후, 그 신경망의 일부 층(하위 층)을 재사용하는 것이 바람직함

## ✓ 효과

- 훈련 **속도** ↑
- 필요한 **훈련 데이터** ↓
- 특히 **심층 합성곱 신경망**(Convolutional Neural Network; CNN)에서 좋은 성능

# 전이 학습 Transfer learning



# 전이 학습 Transfer learning

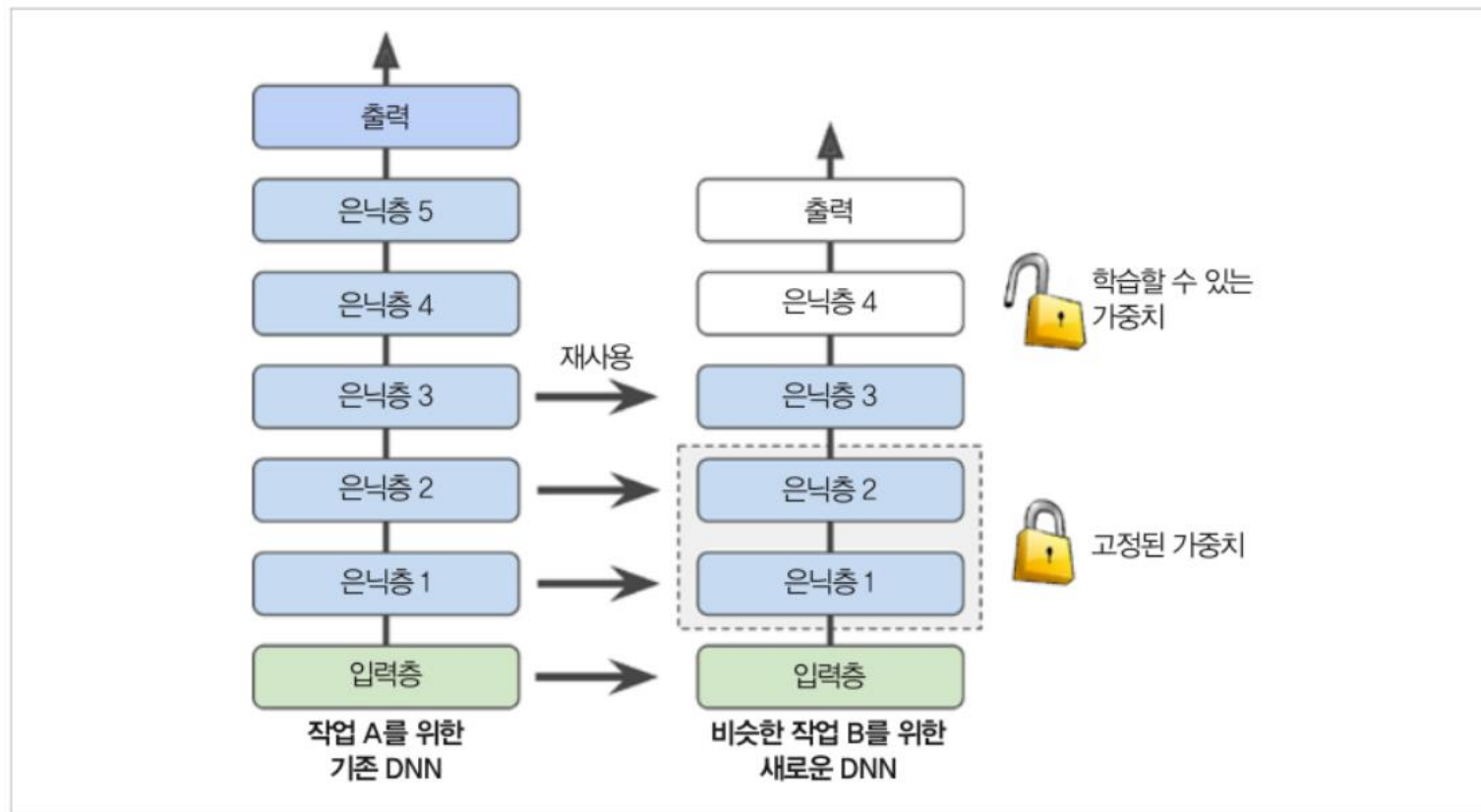


그림 11-4 사전훈련된 층 재사용하기

# 주의 사항

- ✓ 크기가 다른 입력 데이터를 사용한다면,
  - 모델 A (원본 모델): 64 x 64 픽셀 이미지
  - 모델 B (전이 모델): 256 x 256 픽셀 이미지
  - ➔ 모델 B의 입력 데이터(256 x 256 픽셀 이미지)를 64 x 64 픽셀로 변환하는 **전처리** 단계 필요!
- ✓ 원본 모델의 **하위 은닉층**이 더 유용한 정보를 지니고 있음
  - 작업이 비슷할수록 하위 층에서부터 더 많은 층을 재사용(전이) 함
  - 아주 비슷한 작업이면 모든 은닉층을 전이하고 출력층만 새로운 작업에 맞게 교체

# 전이학습 **수행 과정**

## ① 원본 모델 불러오기

```
model_A = keras.models.load_model('my_model_A.h5') # 모델 A 불러오기  
model_A_clone = keras.models.clone_model(model_A) # 모델 A의 구조 복제  
model_A_clone.set_weights(model_A.get_weights()) # 모델 A의 가중치 복사
```

## ② 전이 모델에 은닉층 전이

```
model_B = keras.models.Sequential(model_A_clone.layers[:-1])  
# 모델 B에 모델 A의 출력층을 제외한 입력층과 은닉층 전이  
model_B.add(keras.layers.Dense(1, activation = 'sigmoid'))  
# 모델 B에 출력층 생성
```

# 전이 학습 **수행 과정**

## ③ 전이 층 동결 후, 새로운 층 학습

```
for layer in model_B.layers[: -1]:  
    layer.trainable = False # 모델 B의 전이 층 동결  
model_B.compile(loss = 'binary_crossentropy', metrics = ['accuracy'],  
                optimizer = keras.optimizers.SGD(lr = 1e-2)) # 모델 B 컴파일  
history = model_B.fit(X_train_B, y_train_B, epochs = 4,  
                    validation_data = (X_valid_B, y_valid_B)) # 모델 B 학습
```

## ④ 전이 층 동결 해제 후, 세밀하게 튜닝 (fine tuning)

```
for layer in model_B.layers[: -1]:  
    layer.trainable = True # 모델 B의 전이 층 동결 해제  
model_B.compile(loss = 'binary_crossentropy', metrics = ['accuracy'],  
                optimizer = keras.optimizers.SGD(lr = 1e-4)) # 모델 B 컴파일  
history = model_B.fit(X_train_B, y_train_B, epochs = 16,  
                    validation_data = (X_valid_B, y_valid_B)) # 모델 B 학습
```