



# 사전 훈련된 층 재사용하기

발표자: 권기호

# 목차

A table of contents

1

층의 재사용

2

케라스를 사용한 전이 학습

3

비지도 사전훈련

4

보조 작업에서 사전훈련





1

# 층의 재사용

재사용의 필요성  
전이학습

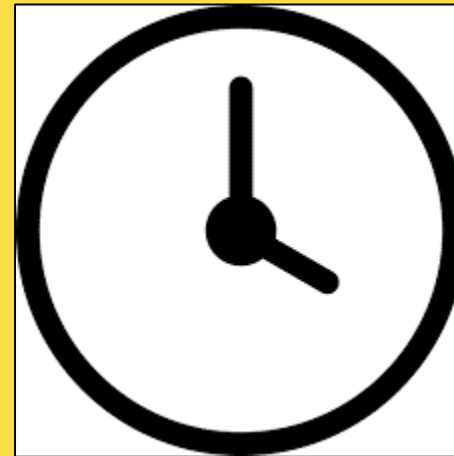
# 1 층의 재사용 필요성

층을 재사용하면 어떤 점이 좋을까요?



## 수많은 데이터의 필요성

큰 규모의 DNN 모델을 스케치(초기) 단계부터 학습하기 위해서는 수많은 (정제된) 데이터가 필요하기 때문에 어려움을 겪는다.



## 긴 학습시간

실무에서 몇 시간 동안 딥러닝 모델을 학습 시킨 후 만족할 만한 결과를 얻을 수 있다. 하지만 매번 딥러닝 모델을 사용하기 위해 몇시간 동안 학습시킬 수 없다.

# 2 전이 학습

비슷한 유형의 문제를 처리한 신경망의 이용

## STEP 1

이미지를 분류하는  
DNN

>>

## STEP 2

구체적인 자동차의  
종류를 분류하는  
DNN 훈련

>>

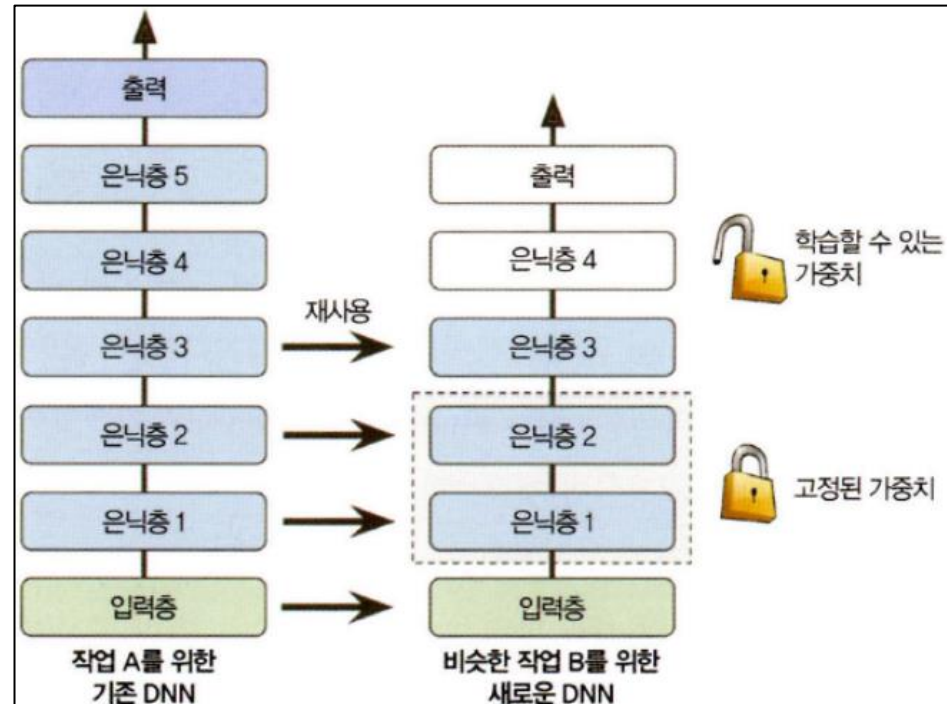
## STEP 3

재사용 층 선정

>>

## STEP 4

재사용 층 동결(가  
중치 고정)  
모델 훈련 및 평가  
가중치 미세 조정



# 케라스를 사용한 전이 학습

문제정의  
케라스를 이용한 전이 학습  
문제점



2

VS.

## 모델 A

8개의 클래스를 분류하는 다중 분류 모델  
성능이 90%이상으로 훌륭함

## 모델 B

샌들과 셔츠 이미지를 구분하는 이진 분류 모델  
양성 = 셔츠, 음성 = 샌들  
데이터가 200개로 적은 편

## 2 케라스를 이용한 전이 학습

케라스를 이용한 전이 학습의 예

### STEP 1

모델 A를 불러온 후 출력층 제외하고 모든 층을 재사용한다.

```
model_A = keras.models.load_model("my_model_A.h5")
model_B_on_A = keras.models.Sequential(model_A.layers[:-1])
model_B_on_A.add(keras.layers.Dense(1, activation="sigmoid"))
```

### STEP 2

얕은 복사를 방지하기 위해 클론은 진행하고, 재사용된 층을 동결하고 적절한 학습 시간을 부여한다.

```
model_A_clone = keras.models.clone_model(model_A)
model_A_clone.set_weights(model_A.get_weights())
```

```
for layer in model_B_on_A.layers[:-1]:
    layer.trainable = False
```

→ 동결

```
model_B_on_A.compile(loss="binary_crossentropy",
                    optimizer=keras.optimizers.SGD(lr=1e-3),
                    metrics=["accuracy"])
```

### STEP 3

동결을 해제하고 작업 B에 맞게 재사용된 층들을 세밀하게 튜닝한다. 학습률을 낮게 하여 기존 가중치를 지킨다.

```
history = model_B_on_A.fit(X_train_B, y_train_B, epochs=4,
                          validation_data=(X_valid_B, y_valid_B))
```

```
for layer in model_B_on_A.layers[:-1]:
    layer.trainable = True
```

```
model_B_on_A.compile(loss="binary_crossentropy",
                    optimizer=keras.optimizers.SGD(lr=1e-3),
                    metrics=["accuracy"])
history = model_B_on_A.fit(X_train_B, y_train_B, epochs=16,
                          validation_data=(X_valid_B, y_valid_B))
```



99.25%



# 3 문제점

성능이 좋지 않다?!

## 001 >> 낮은 성능

타깃 클래스나 랜덤 초깃값을 바꾸면 성능이 떨어진다. 심지어 성능이 향상되지 않거나 더 나빠지기도 한다.

---

## 002 >> 왜?

전이 학습은 작은 규모의 완전 연결(Fully Connected) 네트워크에서는 잘 작동하지 않기 때문이다. 작은 규모는 패턴 수를 적게 학습한다는 뜻이고, 완전 연결은 특정 패턴을 학습하기 때문이다. 그렇게 때문에 범용성이 적어 다른 작업에 유용하지 않다.

---

## 003 >> 어떤 신경망에 유용할까?

더 일반적인 특성을 감지하는 경향이 있는 심층 합성곱 신경망에서 더 잘 작동한다.



3

# 비지도 사전훈련

문제  
해결방법

# 1 문제

직면할 수 있는 다른 문제

레이블된 훈련 데이터가  
많지 않다

비슷한 작업을 진행한  
모델이 존재하지 않는다

레이블 부여 비용(시간, 돈)이  
많이 소요된다

비지도 사전 훈련!

## 2 해결 방법

### STEP 1

레이블이 없는 데이터 또는 전체 데이터를 사용하여 Autoencoder나 GAN와 같은 비지도 학습을 통해 모델을 학습한다.

### STEP 2

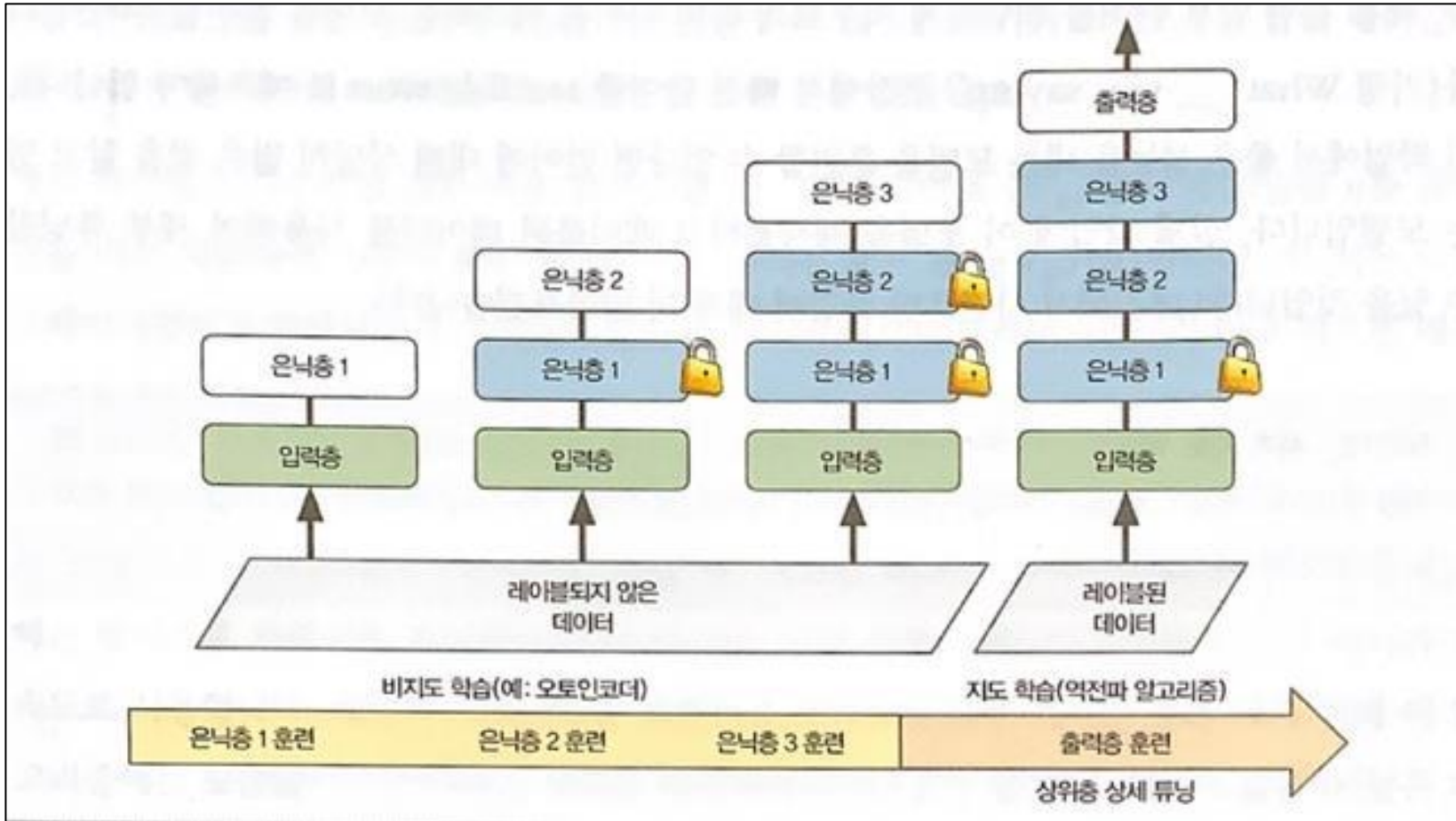
Autoencoder나 GAN 판별자의 하위층을 재사용하고 그 위에 새로운 작업에 맞는 출력층을 추가한다.

### STEP 3

지도 학습 기법을 사용하여 레이블된 데이터를 최종학습을 위해 튜닝한다.

- 딥러닝 초기에는 층이 많은 모델을 훈련하는 것이 어려웠기 때문에 탐욕적 층 단위 사전훈련을 사용했다.
- 하나의 층을 가진 비지도 학습 모델을 RBM을 통해 훈련하고 그 층을 동결해 그 위에 다른 층을 추가하여 다시 모델을 훈련하는 과정을 반복한다.
- 오늘 날에는 훨씬 간단한 방법으로 Autoencoder나 GAN을 통해 한 번에 비지도 학습 모델을 훈련한다.

## 2 해결 방법



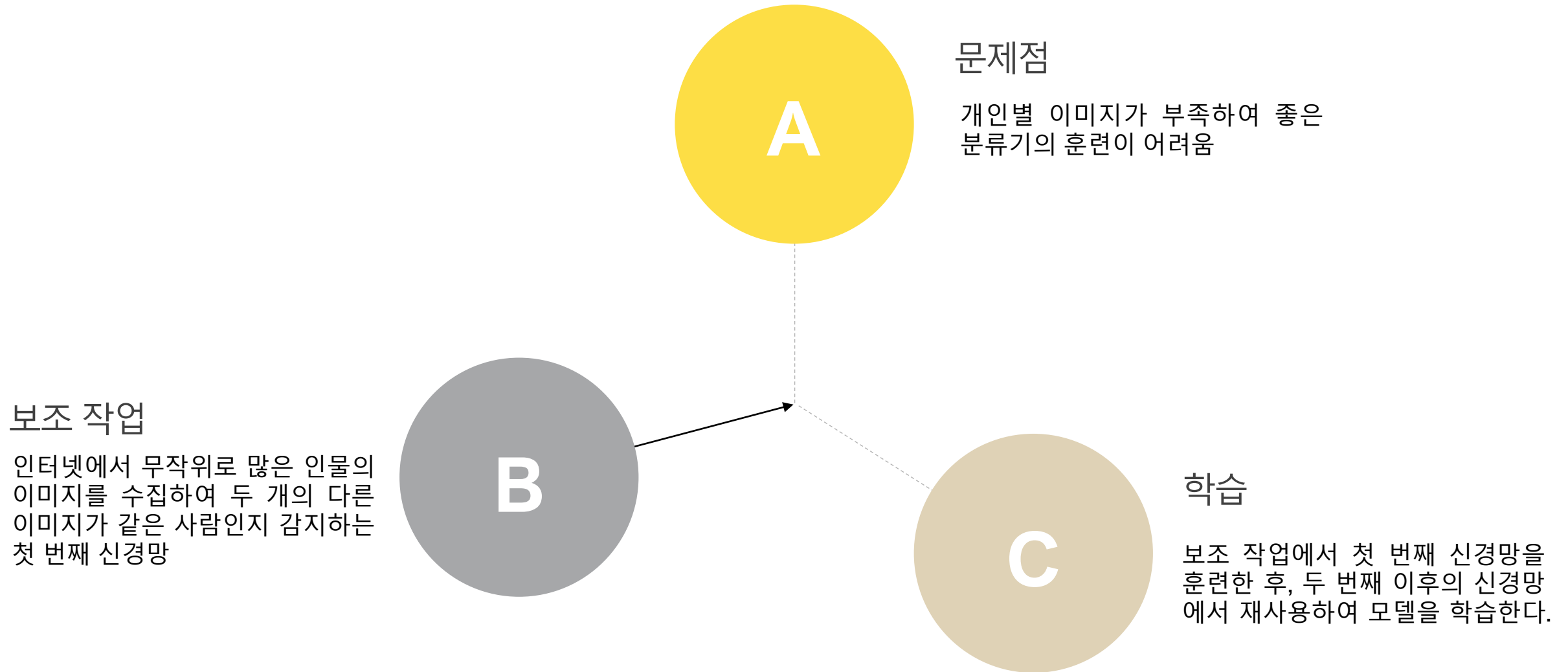
# 4

## 보조 작업에서 사전 훈련

얼굴인식시스템  
NLP

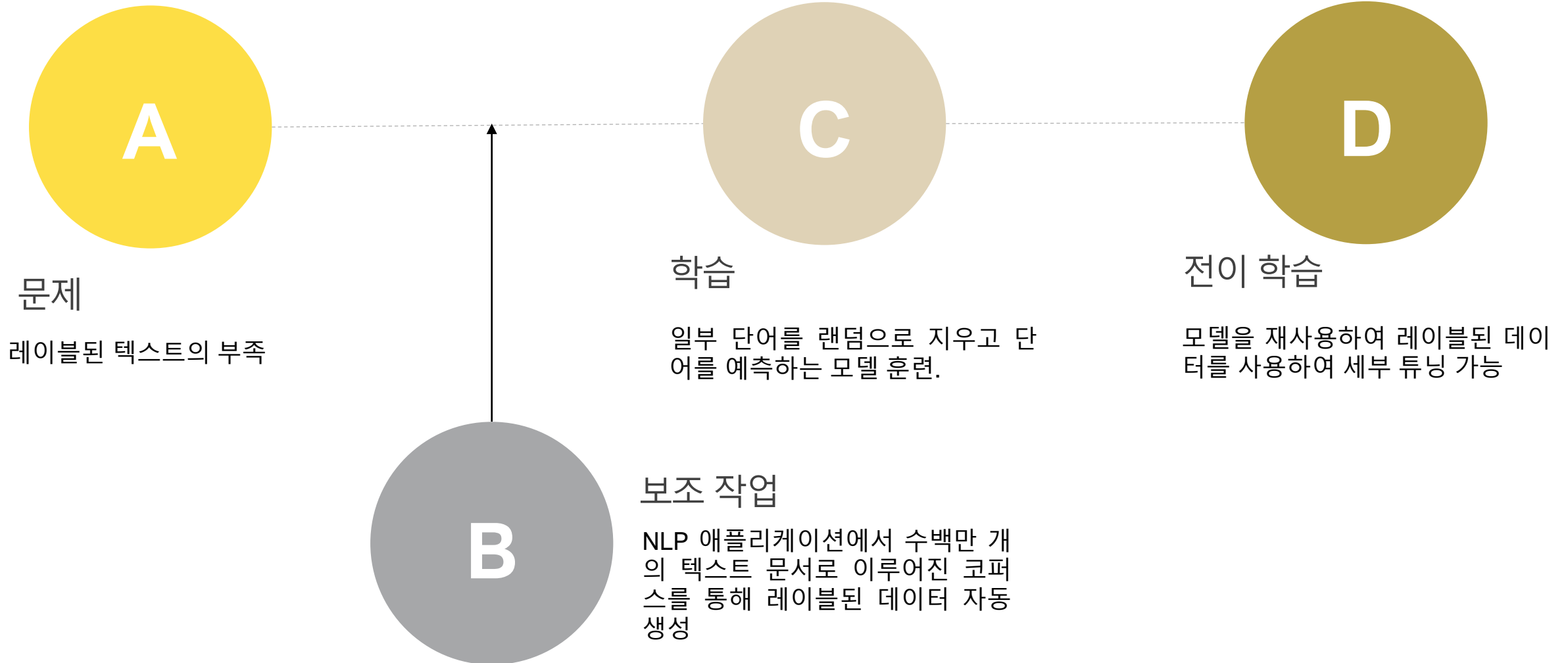
# 1 얼굴 인식 시스템

다른 방법이 있을까?



# 1 NLP

이해력 높은 모델의 재사용





The background of the image is a close-up, top-down view of a cobblestone path. The cobblestones are dark grey or black, with a rough, textured surface. A section of the path, running diagonally from the top right towards the bottom right, is painted a bright yellow. The yellow paint is applied in a way that covers the surface of the cobblestones in that area, though some of the stone texture is still visible. The yellow section is bordered by the dark cobblestones on the left and bottom. In the center of the image, there is a horizontal grey rectangular box containing the Korean text '질문 타임' in yellow.

질문 타임





**Thanks!**