

- 12/19
- 이미지 (Image) 특징을 추출해야요 ~
- DNN을 이용하면 알쁘 M·L 보다 특징을 잘 되요!
- CNN을 이용해서 이미지의 특징을 추출해서 특징하고 예측하면


Filter를 이용해 convolution 연산을 통해 이미지의 특성 추출

이 Filter의 선호도가 학습이 훨씬 빠르고 정연히 개선되며
어느정도는 좋아지만 만족할 만한 선호도 아니예요!!

※※ Augmentation(증식)을 통해 일부 학습 데이터가
이미지 개수 많이 부족 (결과적으로 Overfitting이 발생) 가능!!

이미지 파일을 처리할 때 handling 해야해요
※※ Image Data Generator를 적용
※※ 속도가 느리다는 단점

Tensorflow 2

- Tensorflow는 전용파일이 있어요 TFRecord'라고 불리는 Tensorflow 전용 데이터포맷이 있어요!
- 이미지 흐름이 어느정도는 전환될 수 있어요!!
- 여기서 더 좋은 모델을 만들려면 어떻게 해야 합니까?

좋은 이미지 특성을 이용하기 모델을 훈련하면 돼요!!

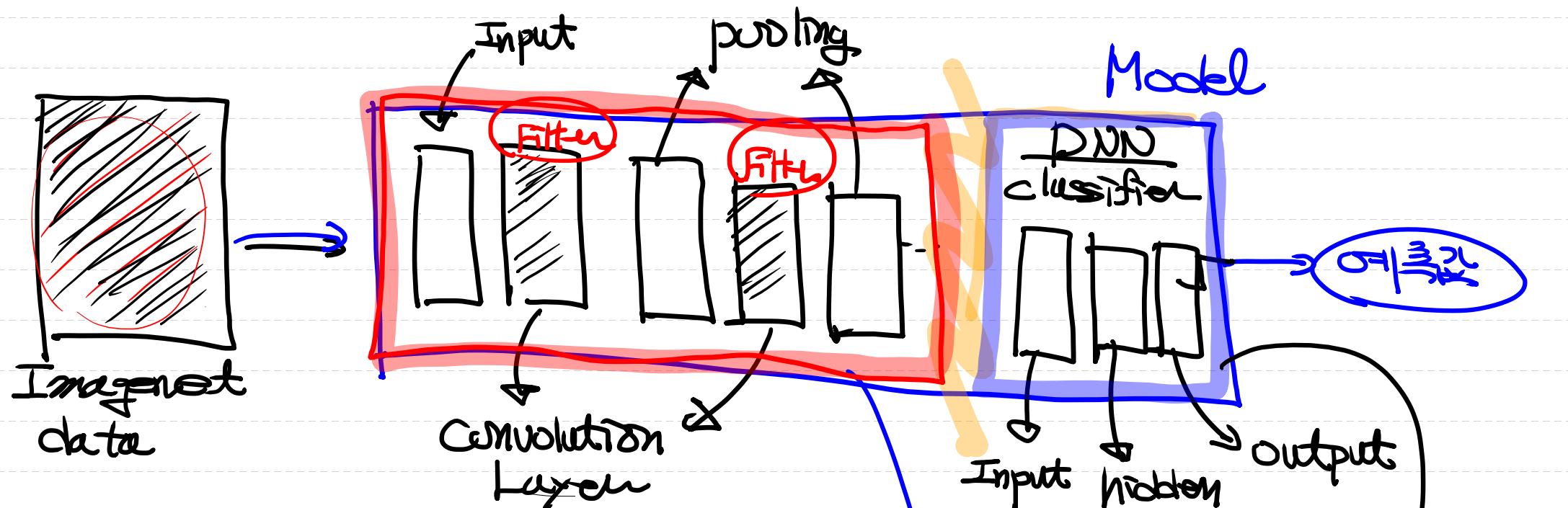
Filter가 좋아야 해요!!

Google
MS

→ 아주 큰 CNN Model을 써요.

→ 사용하는 공개된
Image data set

Coco dataset, pascal VOC dataset, ImageNet ~ Class 1000



✓ Inception (구글) → 32개

✓ ResNet (MS) → 152개

✓ MobileNet

✓ EfficientNet

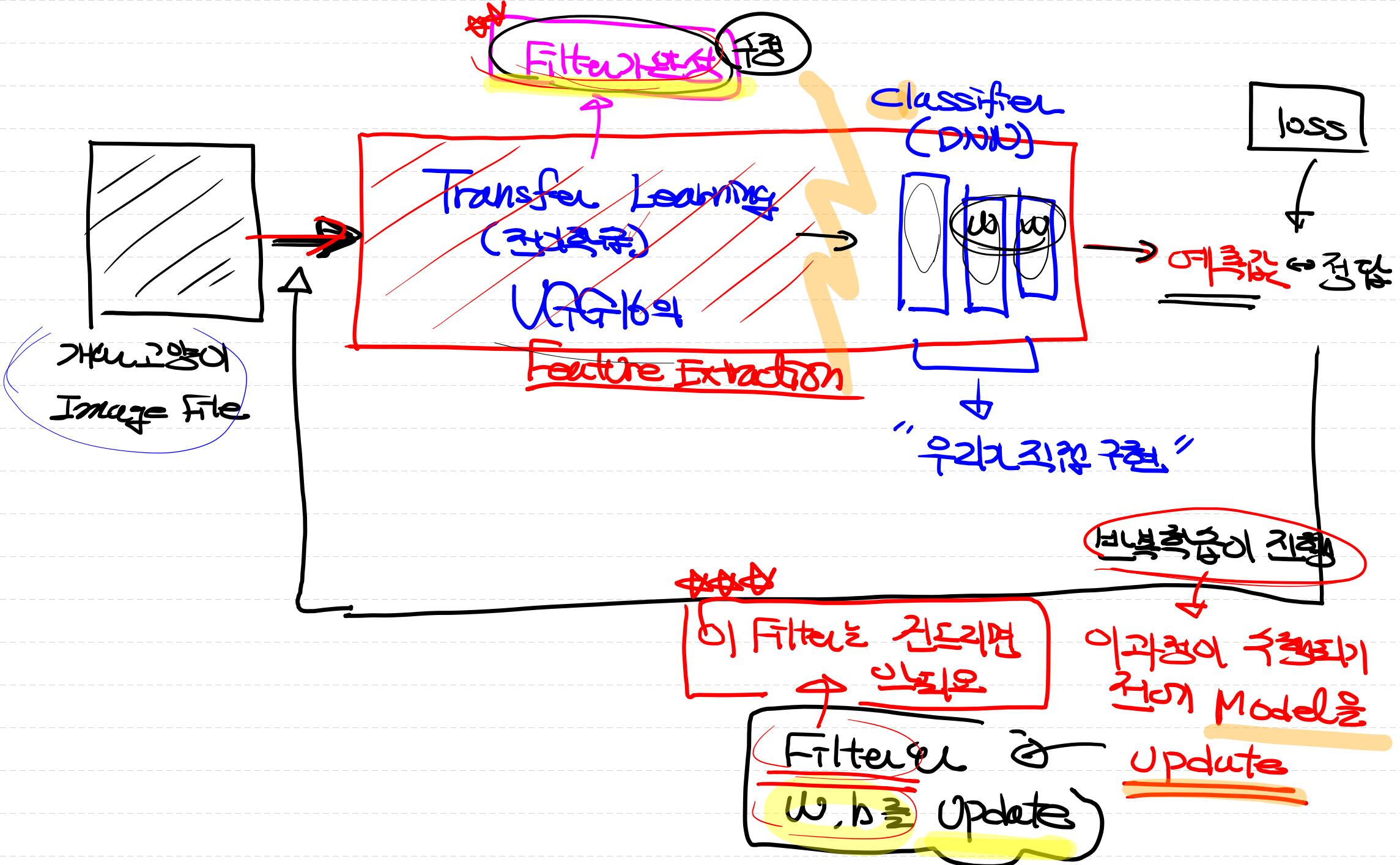
• VGG16, 19 → 예측용
...
Filter

입력으로 들어온
이미지의 특징을
추출하는 역할

1000 개의 이미지
분류를 학습하고
예측하는거

사실상 일부분을
가져와서 사용해
는겁니다!!

Imagenet data
가지고 만들 Filter



Transfer Learning (전이학습) → 일반화의 방법!!

트랜스퍼학습은 뇌가 놓아요!!

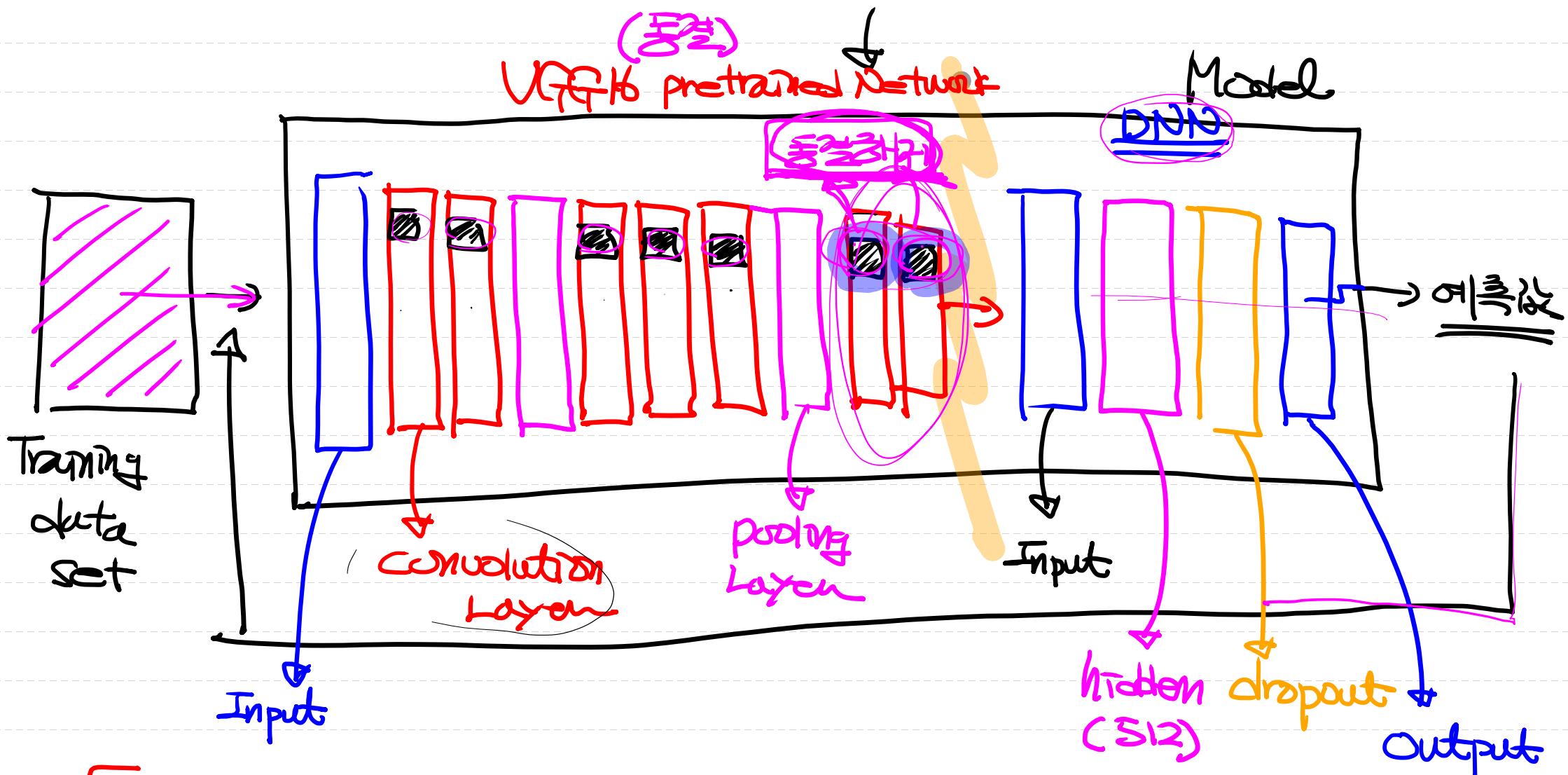
<Fine Tuning> → "마세조정"

만약 Transfer Learning으로 모델을 학습했어요!!

여기서 추가적으로 우리 모델을 조정하는걸 → Fine Tuning

모델 학습이 끝난후 설정을 조금 바꾸면 재학습 진행!!

이전 학습된 Model을 인식화 후



기본 학습을 진행 (특정 convolution Layer의 풍경을 풀고)

다시 학습을 진행

→ Fine Tuning

- Vision data 예측
(0)마지)

① 일반 M-L기법 (Regression) 성능이 아주 좋지 않아요!

② DNN (Deep Neural Network) → 예측을 가능합니다

성능은 70%

③ CNN (Convolutional Neural Network) 80%
→ 예측도 가능하고 성능도 나쁘지 않아요!

data (3250장의 data)

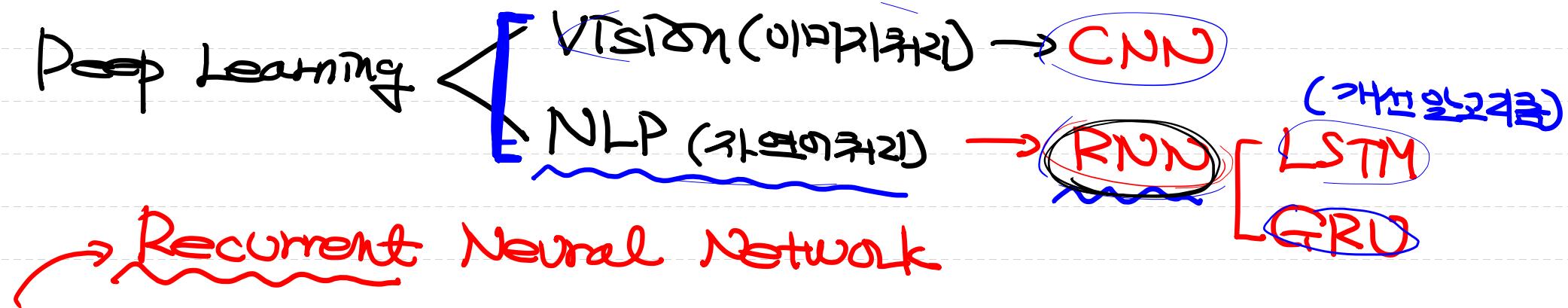
④ Transfer Learning (Pre-trained Network)

VGG16 model을 이용 → 성능이 92%

✓ ⑤ Fine tuning 기법으로 예측률 → 성능이 94%.

✓ ⑥ EfficientNet을 pre-trained Network 사용 → 98~99%

Deep Learning의 포화를 뛰어넘고 있는 RNN



RNN은 기본적으로 sequential data를 학습할 때 사용되는
(초기 있는 드라이브) 일고리즘

"time series data" (시계열 데이터) → Text

연관성

“내가 놀고 있던 이전보다 잘 만들었나?”
(아주 재미있었습니다)

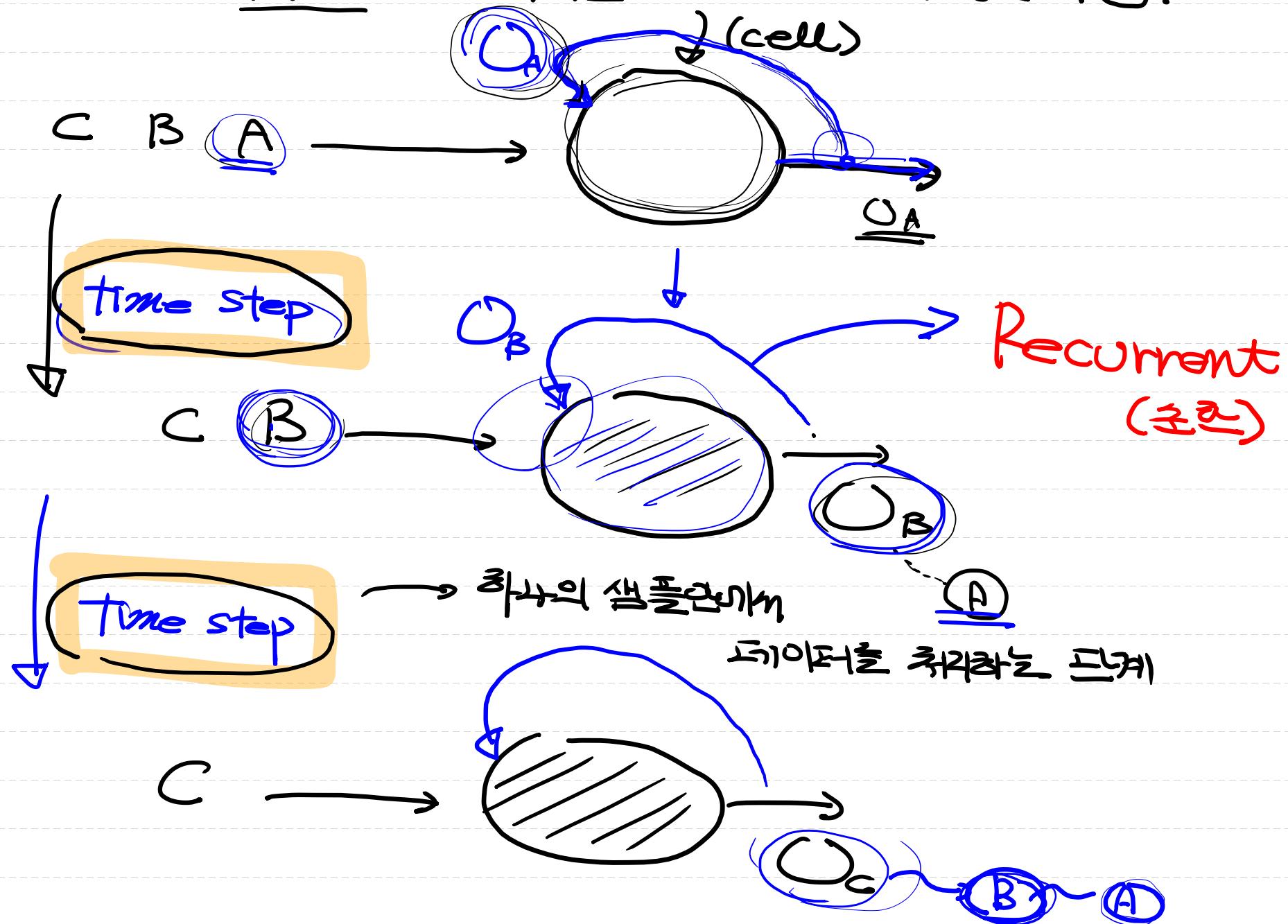
→ I am a boy (o)
부정 I buy a I am (x)
정정

(A, B, C)

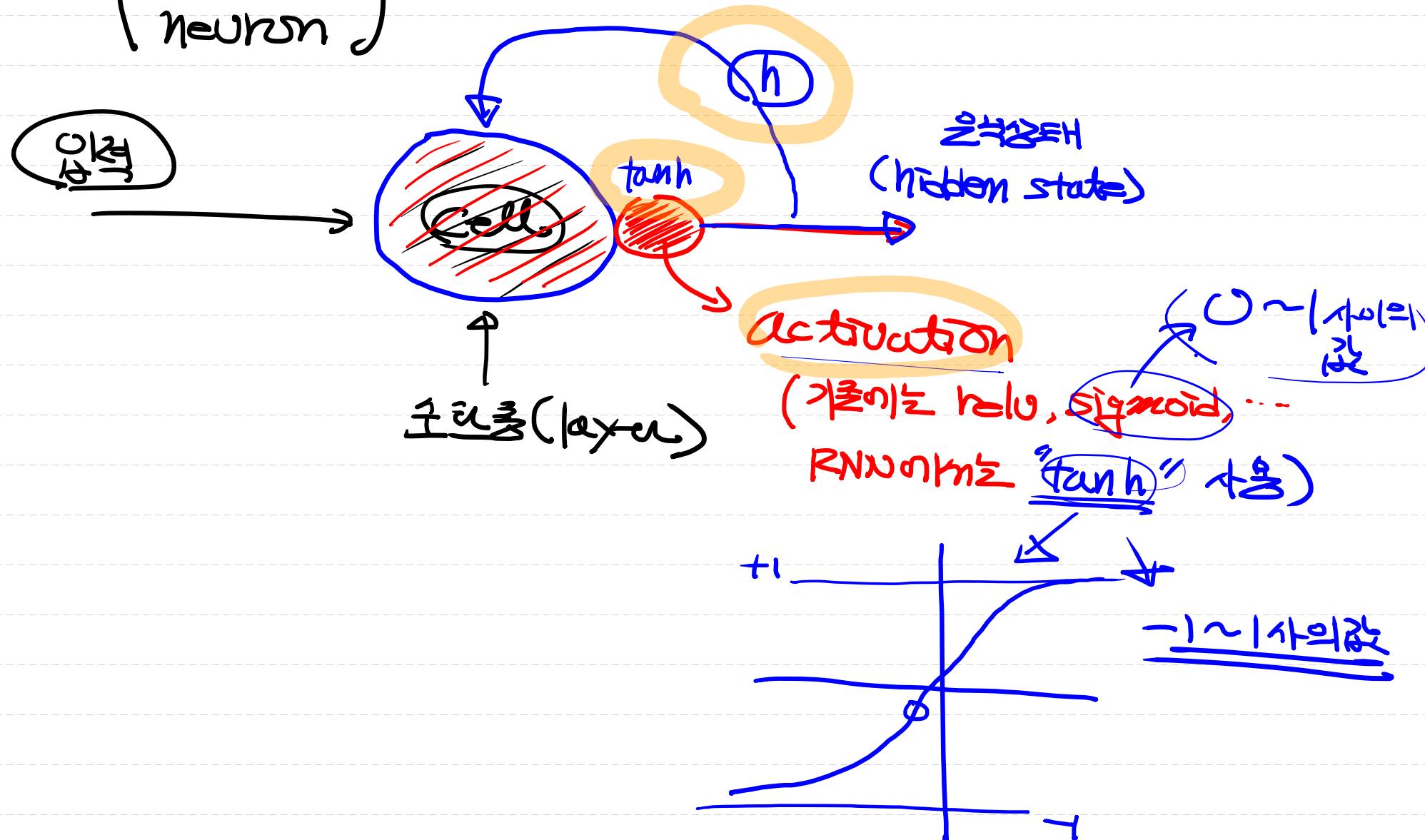
예를들면)

하나의 sample만에 3개의 data가 있어요

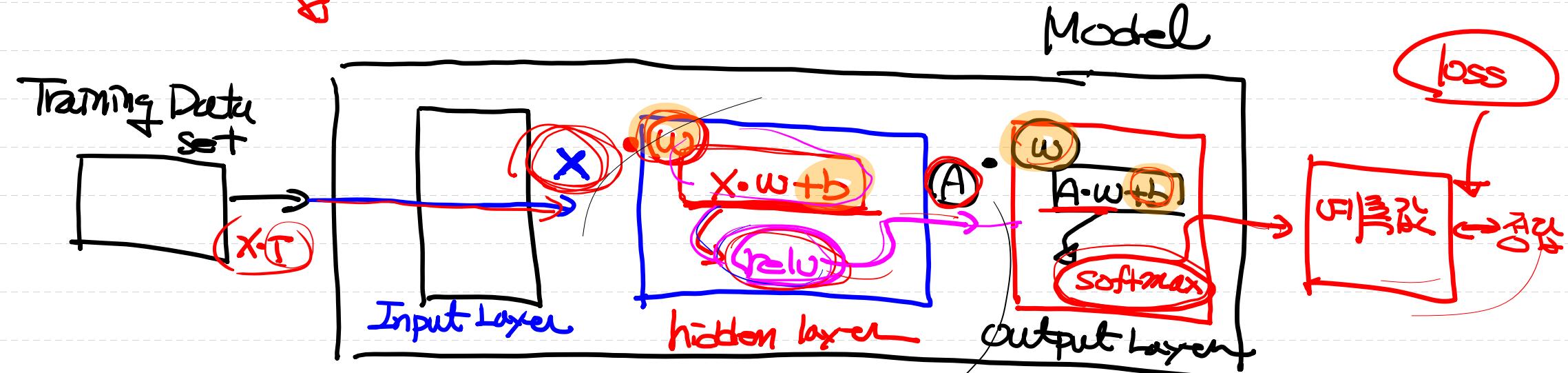
RNN으로 처리하는 neuron이 있다고 가정.



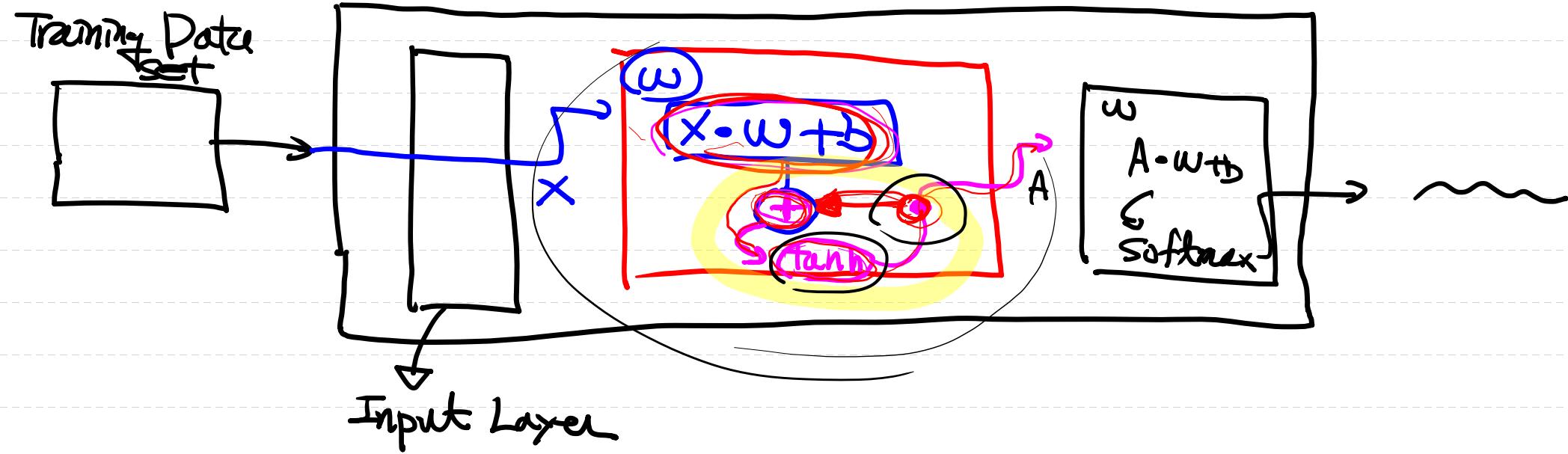
RNN^o (layer) $\stackrel{\text{neuron}}{=} \text{cell}$ 이라고 표기해요!



④ DNN과 RNN의 차이를 그림을 통해 알아보아요 !!



RNN



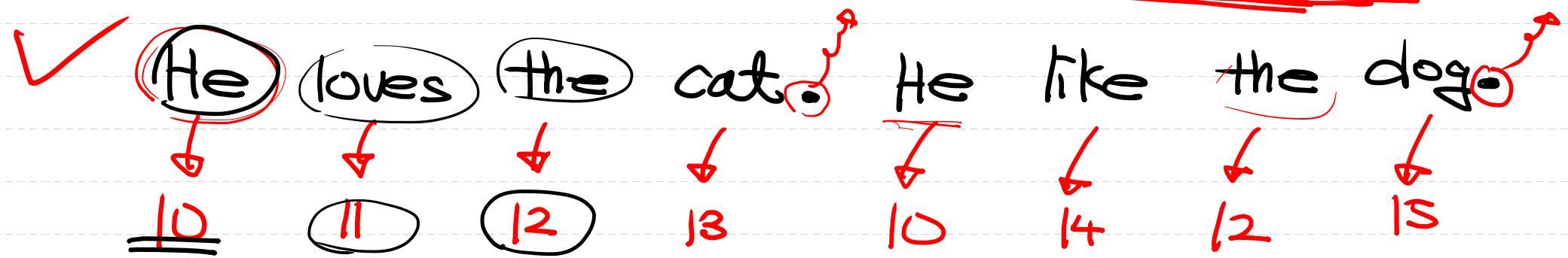
① 대표적인 RNN 예제로 RNN을 이해해보아요 ~

→ IMDB review dataset → 이미 글자로 변경된 데이터가 제공.

영화평론은 시청평론에는 숫자만 들어온다 였어요 (글자)에 포함)

Text data를 글자로 변환해야 알맞아야 해요 !!

각 문장을 문장으로 데이터로 변경하고 데이터에 고유한 정수를 부여해요 !



→ 단어를 하나의 정수에 mapping하고 동일한 단어는 동일한 값을
정수에 mapping. → 모두 소문자로 변경하고 period (- 마침표)

값은 지우하고 굵백으로 단어를 보기. → Token

• 영어는 의의 과정이 풍자적으로 간단하다

한국어의 경우 이 과정이 어렵다 (단락의 구조로 분리 X)

→ 형태소 분석이 필요 → (library로 해결 → KoNLPy)

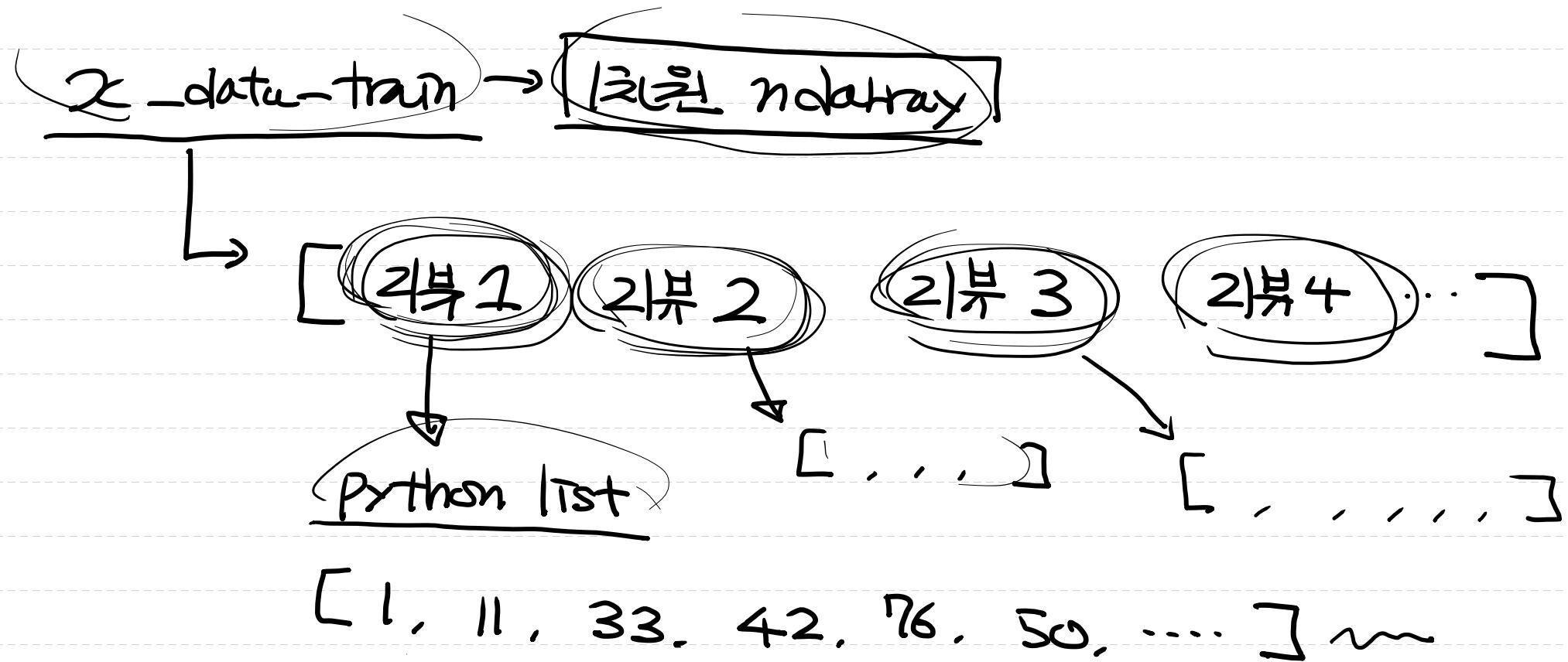
결국, training data set의 고유한 단어를 뽑아서 풍자로 만들고

있어요 → 이를 vocabulary (어휘사전, 명사사전)이라고 해요.

Vocabulary 안에는 tokens가 들어가 있어요 → token마다 고유한
정수를 mapping

특수한 정수값들이 있어요

- 0 : padding
- 1 : 문장의 시작
- 2 : 어휘사전에 없는 Token



Review는 글주제하는 사람마다 길이가 각각 달라요

↳ 2차원 ndarray로 전체 데이터를 표출하면 굽은 경비가 생기요

