




텐서플로 기초와 모델 소용-20학번-김희성

미래자동차 로봇 트랙

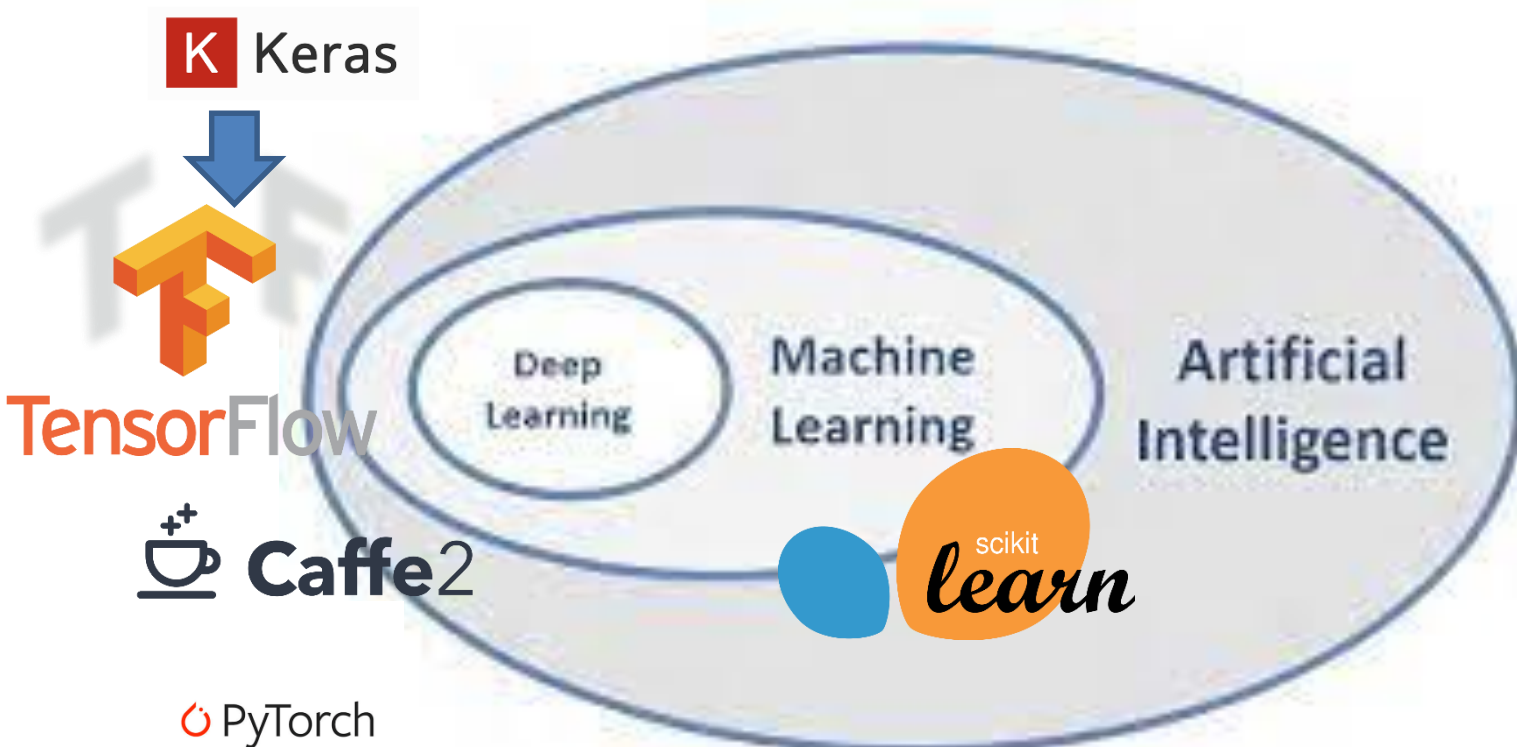
Yesform.com contribute not only to the forms & documents
but also success vision through business consultation and
the whole heart companionship.

CONTENTS

A decorative graphic consisting of a light green horizontal bar, a small light green circle, a long purple horizontal bar, and three purple circles of varying sizes.

- 
- A decorative graphic consisting of a long purple horizontal bar and two purple circles of varying sizes.
- 01 딥러닝을 위한 여러가지 framwork
Tensorflow, karas, pytorch, caffee...
 - 02 책 내용 요약
Tf1 vs tf2
Keras, model fitting
 - 03 머신러닝 과정
preprocessing, crossvalidation,
 - 04 퀴즈 풀이
Quiz.
 - 05 하다가 궁금했던 점.
마지막 요약.

딥러닝을 위한 여러가지 framework



딥러닝을 위한 여러가지 framework

Table of Contents



Top Deep Learning Frameworks

1. TensorFlow
2. PyTorch
3. Keras
4. Sonnet
5. MXNet
6. Swift for TensorFlow
7. Gluon
8. DL4J
9. ONNX
10. Chainer

Wrapping Up

Tensorflow

- 구글이 만듦.
- 가장 인기 있는 deep learning framework(library)
- 병렬처리, GPU 사용가능.

01

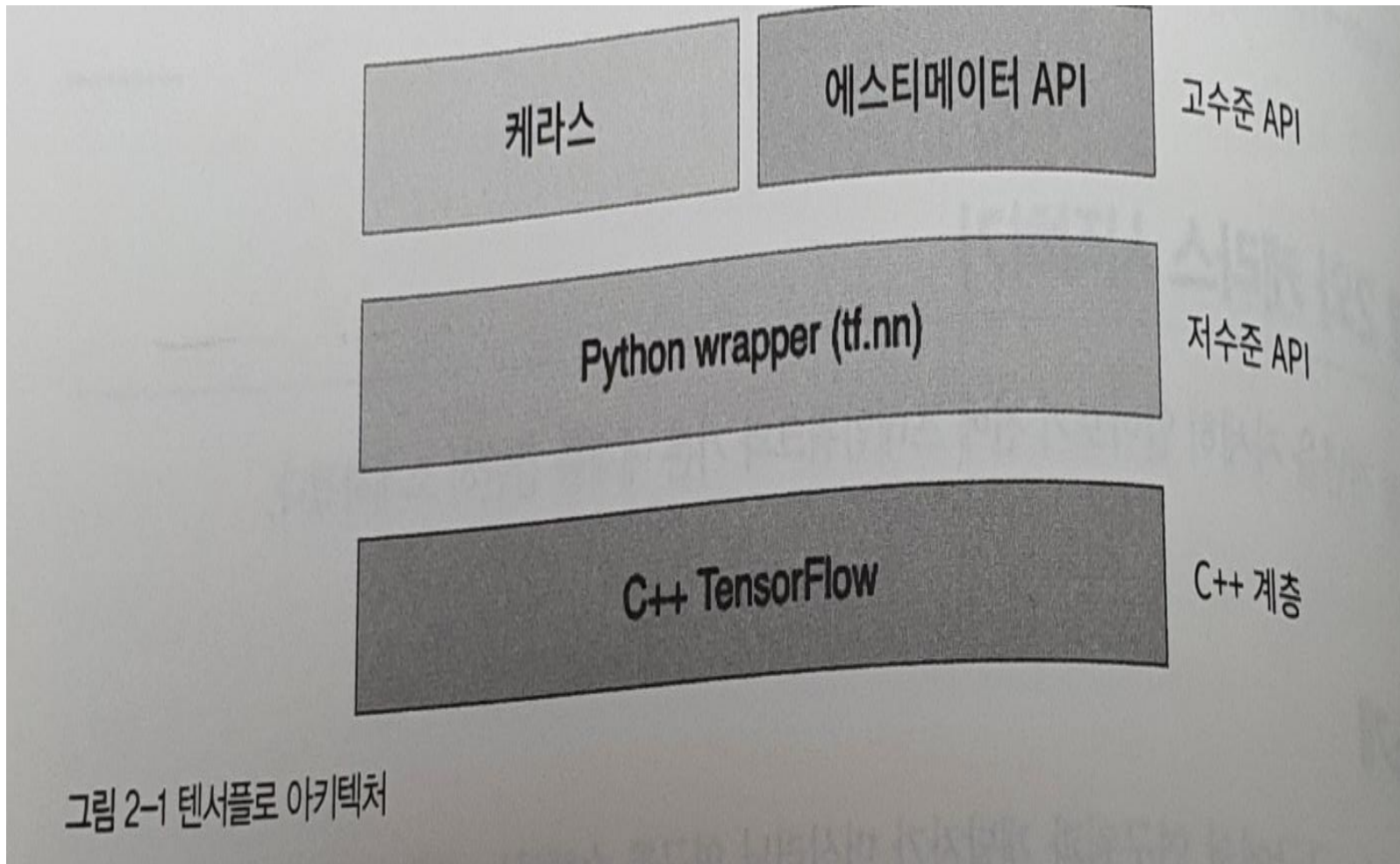
책 내용 소개

Yesform.com vision is to become a good partner with all business.

1. text API 소개
2. 모델 구성 (순차형 vs 함수형)
3. 텐서
4. 그래프와 최적화 (오토 그래프)
5. 그래디언트 테이프
6. 콜백
7. estimator
8. Tensorflow extened, tensorflow addon, tensor board.

책 내용 요약

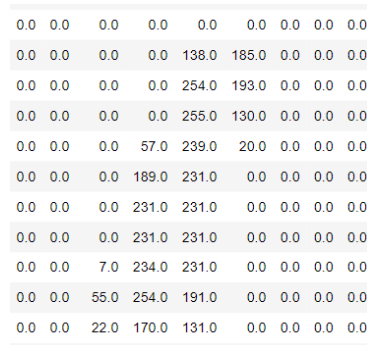
Api 사진



순차형 모델 VS 함수형 모델

flatten

Output layer

[illegible]

책 내용 요약

순차형 모델 VS 함수형 모델

순차형

tensorflow.keras.models.
sequential

Add method

flatten

Hidden
layer

output
layer

함수형

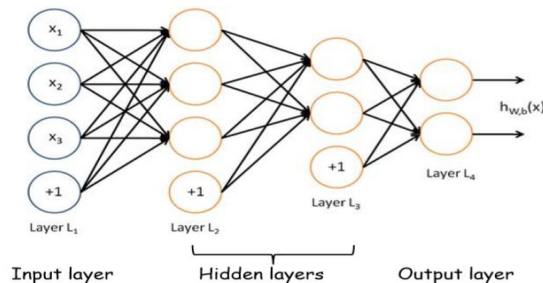
flatten

Hidden
layer

Output
layer

각각 구성 해
연결

세세한 조정
가능



책 내용 요약

이렇게 만들어진 model

.summary() => 상태 파악

.get_weight => 각 eapoach 별로 weigh이 나옴.

.fit() -> training 시킴. Validation accuracy 기능까지 탑재.

.layers => 모델 계층과 형상 목록

```
Epoch 19/25  
60000/60000 [=====] - 2s 33us/sample - loss: 0.1095 - accuracy: 0.9697 - val_loss: 0.1172 - val_accuracy: 0.9648  
Epoch 20/25  
60000/60000 [=====] - 2s 33us/sample - loss: 0.1053 - accuracy: 0.9713 - val_loss: 0.1153 - val_accuracy: 0.9663  
Epoch 21/25  
60000/60000 [=====] - 2s 33us/sample - loss: 0.1017 - accuracy: 0.9717 - val_loss: 0.1128 - val_accuracy: 0.9670  
Epoch 22/25  
60000/60000 [=====] - 2s 32us/sample - loss: 0.0983 - accuracy: 0.9725 - val_loss: 0.1095 - val_accuracy: 0.9681  
Epoch 23/25  
60000/60000 [=====] - 2s 33us/sample - loss: 0.0949 - accuracy: 0.9735 - val_loss: 0.1066 - val_accuracy: 0.9693  
Epoch 24/25  
60000/60000 [=====] - 2s 33us/sample - loss: 0.0917 - accuracy: 0.9747 - val_loss: 0.1054 - val_accuracy: 0.9695  
Epoch 25/25  
60000/60000 [=====] - 2s 33us/sample - loss: 0.0891 - accuracy: 0.9758 - val_loss: 0.1040 - val_accuracy: 0.9696
```

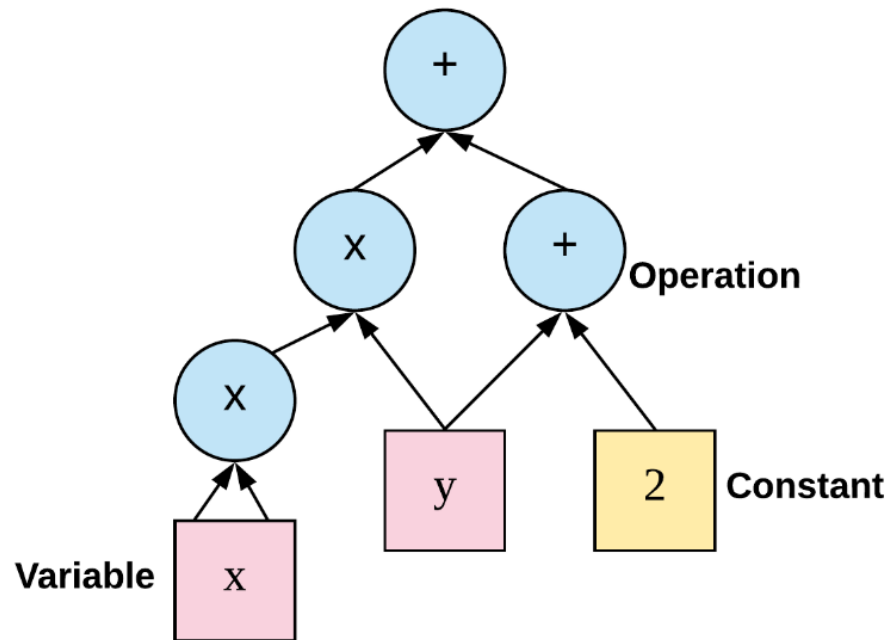
Layer (type)	Output Shape	Param #
flatten_19 (Flatten)	(None, 784)	0
dense_38 (Dense)	(None, 128)	100480
dense_39 (Dense)	(None, 10)	1290
Total params: 101,770		
Trainable params: 101,770		
Non-trainable params: 0		

우리가 각 계층을 만들고,
정확도를 확인할 수 있구
나.

책 내용 요약

Tf에서는 위 과정들을 텐서와 그래프로 나타냄.

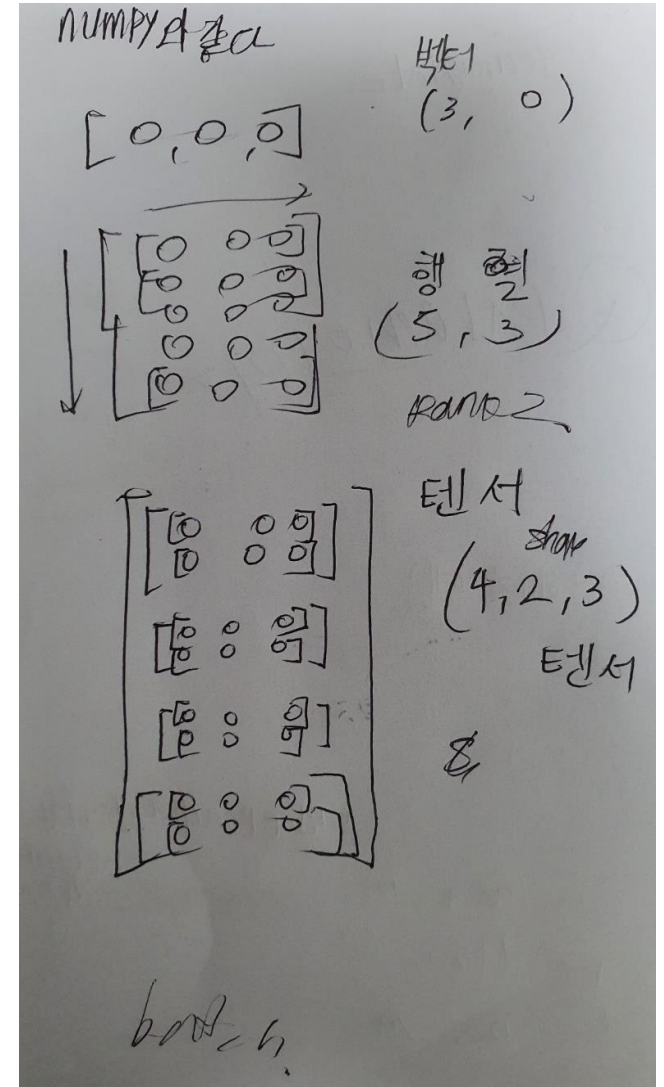
- 병렬 처리나 최적화에 좋다.



책 내용 요약

텐서

- CONST VS VARIABLE
- TYPE (INT8, STRING, FLOAT16, FLOAT32)
- SHAPE (텐서의 모양)



책 내용 요약

그래프와 최적화. (오토 그래프)

- Tensorflow 1 vs Tensorflow 2
- Lazy execution vs eager execution
- 버전 1에서는 최적화 문제 때문에 한번에 연산이 되지 않았지만, 버전 2에서는 한번에 계산 가능
- Graph optimizer -> tf. Function decorator
- 다른 기기로 보내야할 때
- 최적화 필요할 때.

```
@tf.function
def compute(a,b,c):
    d = a*b + c
    e = a * b * c
    return d,e
compute(a,b,c)
```

책 내용 요약

Gradient tape

<자동 미분 수행 가능> 자동으로 모든 연산을 기록 가능.

```
: 1 A, B = tf.constant(3.), tf.constant(6.)
  2 X = tf.Variable(20.)
  3 loss = tf.math.abs(A * X - B)
  4 def train_step():
  5     with tf.GradientTape() as tape:
  6         loss = tf.math.abs(A * X - B)
  7         dX = tape.gradient(loss, X)
  8         print("x = {:2f}, dx = {:2f}".format(X.numpy(), dX))
  9         X.assign(X - dX)
 10
 11 for i in range(7):
 12     train_step()
```

```
x = 20.000000, dx = 3.000000
x = 17.000000, dx = 3.000000
x = 14.000000, dx = 3.000000
x = 11.000000, dx = 3.000000
x = 8.000000, dx = 3.000000
x = 5.000000, dx = 3.000000
x = 2.000000, dx = 0.000000
```

책 내용 요약



Keras 콜백

모델을 fitting 할 때 , 쓸 수 있는 함수

CSVLogger : 훈련정보를 csv file로

EarlyStopping : 더 이상 개선되지 않으면 훈련 중지. 과적합을 피함.

LearningRateScheduler : 세대마다 학습률 변동.

이런 것이 있구나.



책 내용 요약



Estimator : 훈련 평가 예측 을 단순히 하는 것.

사전 제작된 에스티메이터

DNN, linear regressor 등이 있다.

맞춤형 에스티메이터

keras model을 estimator로 변환함.

이런 것이 있구나.



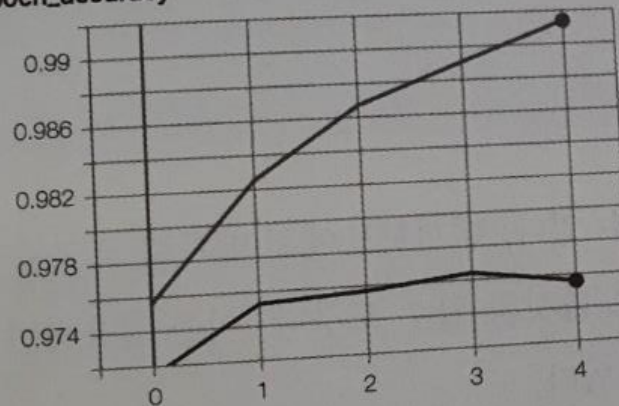
책 내용 요약

Tensorflow extended
Tensorflow addons

을 이용해 추가적인 활동가능

****텐서 보드**** : 파라미터별 시각화 도구

epoch_accuracy



epoch_loss

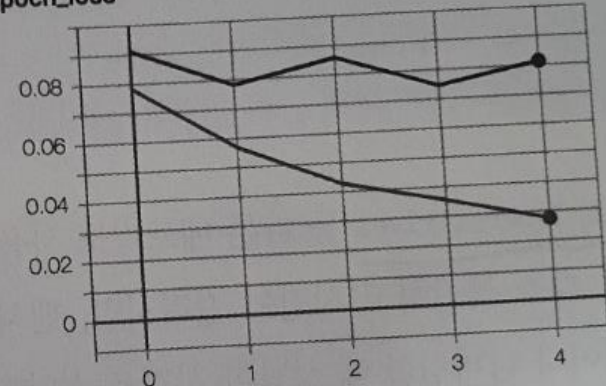


그림 2-5 훈련 동안 텐서보드에서 표시하는 두 개의 그래프

퀴즈 풀이

0. 텐서플로와 비교했을 때 케라스는 무엇이며 목적은 무엇인가?

>> 케라스는 다른 모듈로 존재하기도 하지만, 2017년 이후, tensorflow에서 사용자가 쉽게 사용하기 위한 api중 하나.

1. 텐서플로가 그래프를 사용하는 이유와 수동으로 그래프를 생성하는 방법은 무엇인가?

>> 최적화와 분산처리 병렬처리, GPU 연산에 유리하다.

2. 조급한 실행모드와 느긋한 실행 모드의 차이점은 무엇인가?

>> 계산이 정의 단계에서 바로 되느냐 안 되느냐 차이.

```
@tf.function
def compute(a,b,c):
    d = a*b + c
    e = a * b * c
    return d,e
compute(a,b,c)
```

3. 텐서보드에 정보를 기록하고 그것을 표시하는 방법은 무엇인가?

>> 텐서보드의 내용은 tf.summary에 저장된다.

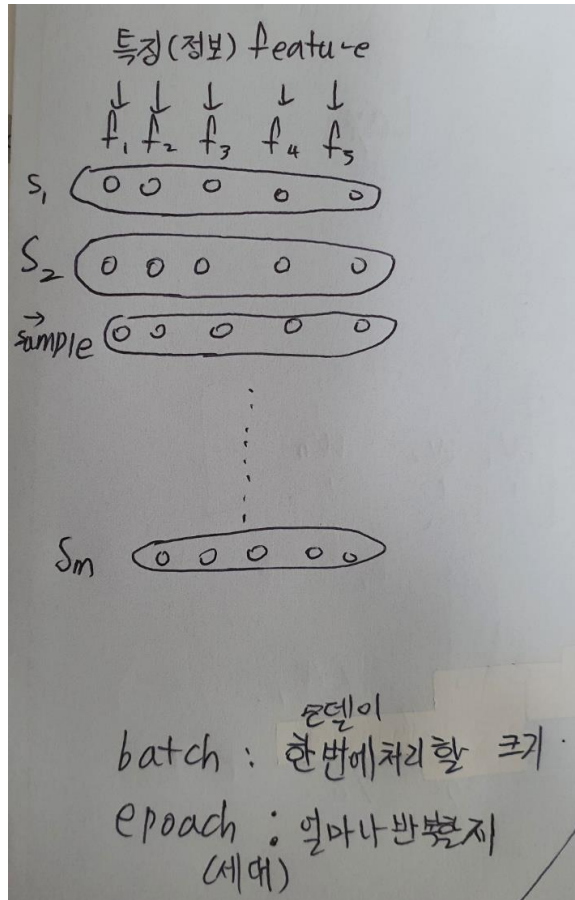
Tensorboard -logdir ./logs_keras 로 시각화하여 볼 수 있다.

4. 텐서플로 1과 텐서플로 2의 주요 차이점은 무엇인가?

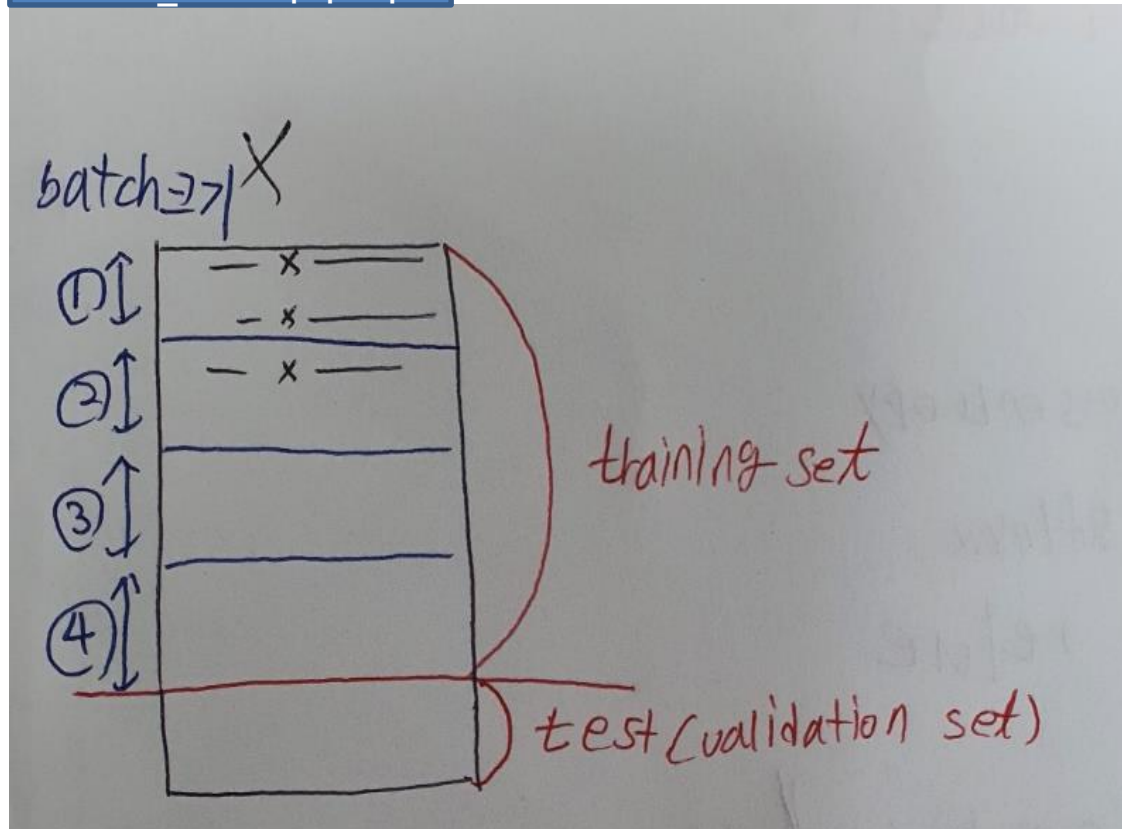
>> 가장 큰 차이는 느긋한 실행 모드와 조급한 실행 모드

ANN 과정 요약.

2. Data 구조

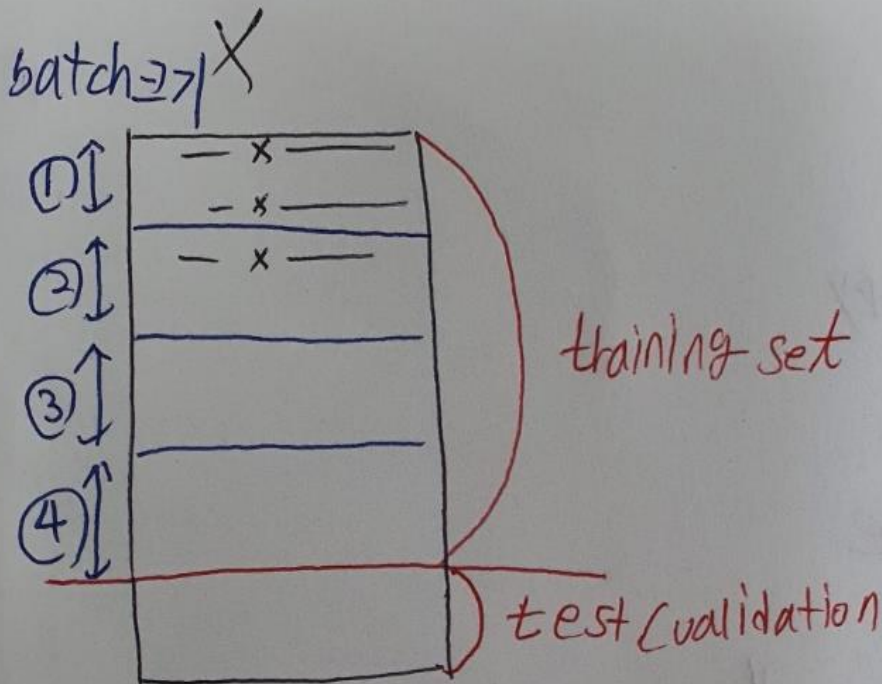


2. test_set train_set 나누기



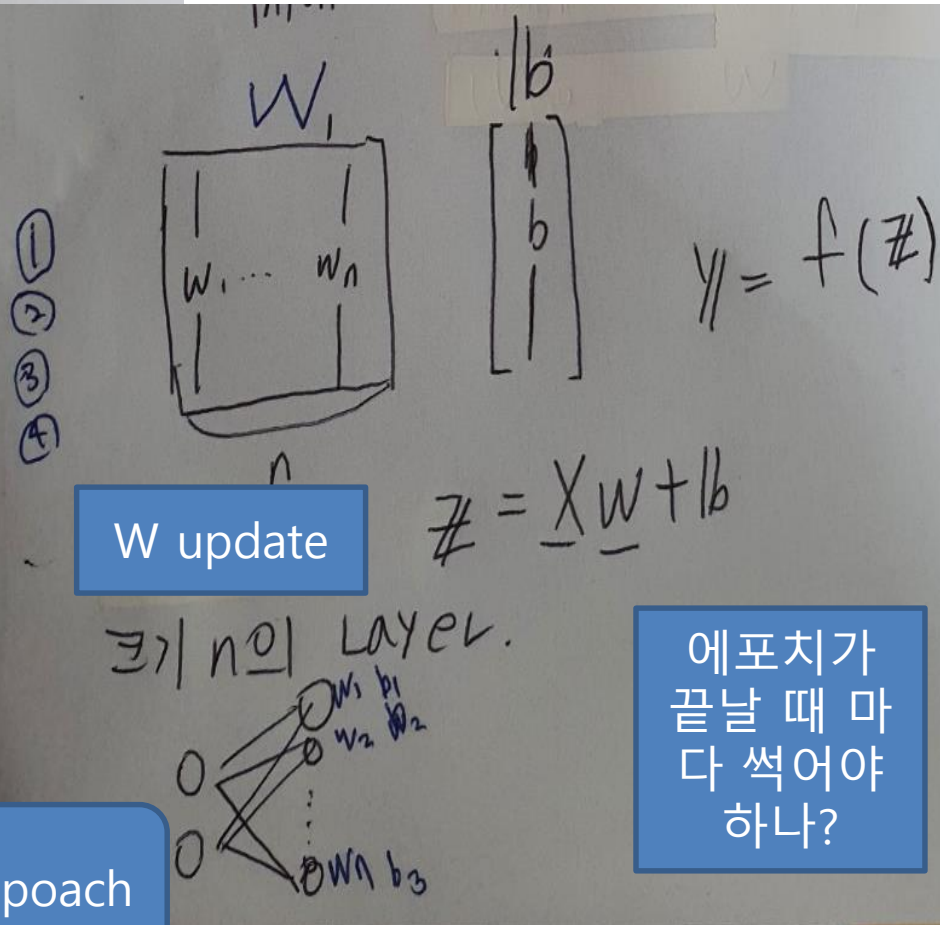
ANN 과정 요약.

2. test_set
train_set 나누기



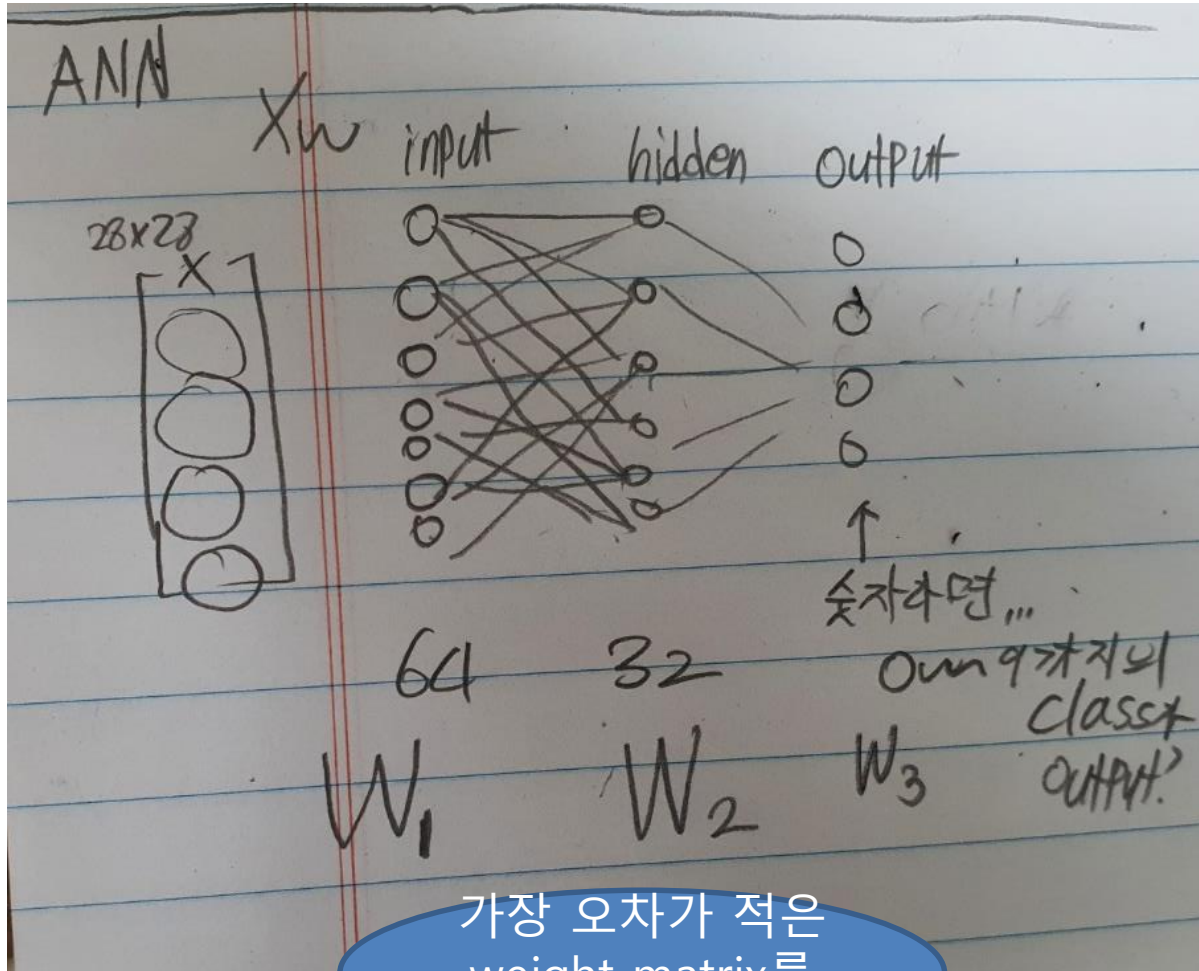
3. 모델 훈련

처음에는 정규분포



Batch vs epoch

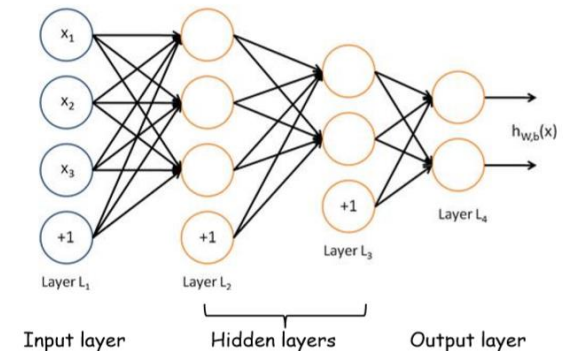
ANN 과정 요약.



가장 오차가 적은
weight matrix를
찾는 과정

ANN
(artificial neural network)
부터 이해가 잘 안감.

Output layer 가 logistic
regression으로, 각 class
가 속할 확률?



궁금한 점.



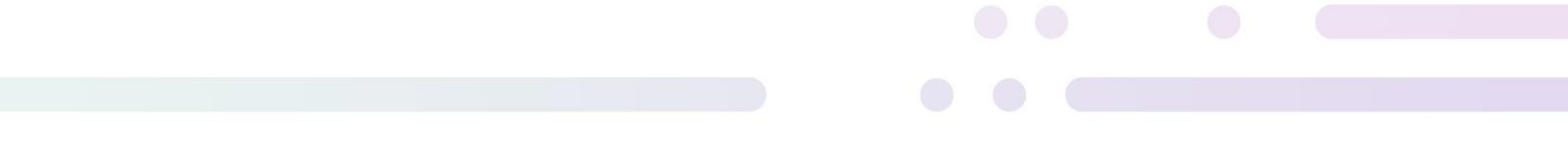
1. 아직 알고리즘이 완벽하게 이해되지 않는데, 전부 이해하고 넘어가는 것이 맞는가? 아니면 활용에 초점을 맞추는 것이 맞는가?
2. Loss function을 Cross entropy로 쓰는 이유가 무엇인가?
3. Activate functio의 선형성? 이 무엇인가? ReLU는 어떻게 나온 것인가?
4. ANN의 output이, 어떻게 구성되는지 잘 모르겠음.
5. ANN이 regression에 쓰이는 방법이 이해가 가지 않음.

더 찾아 볼 것...



발표 - 요약



0. 다양한 딥러닝 프레임워크가 있고, 우리는 tensorflow를 사용한다.
 1. 순차형과 함수형으로 모델을 만들고 훈련시킬 수 있다.
 2. 텐서 플로우에서는 텐서와 그래프를 활용한다. (오토 그래프로 자동 최적화 가능)
 3. Gradient tape로 자동 미분 연산 가능
 4. Keras callback으로 overfitting을 막거나, log를 기록하는 기능이 있음.
 5. 텐서 보드를 이용한 시각화 가능.
- 

THANK YOU

Yesform.com contribute not only to the forms & documents
but also success vision through business consultation and
the whole heart companionship.

