

13기 정규세션

ToBig's 12기 이승현

실습과 함께하는

딥러닝 프레임 워크

Contents

Unit 01 | Intro

Unit 02 | Overview – Tensorflow VS Pytorch

Unit 03 | Tensorflow Tutorial

Unit 04 | Pytorch Tutorial

Unit 05 | Assignment

Unit 01 | Intro



이제 Numpy로 Back Propagation 구현 안해도 됩니다!
우리엔 프레임워크가 있으니까요!

Unit 01 | Intro

프레임워크란 무엇일까요?

응용 프로그램을 개발하기 위한 여러 라이브러리나 모듈 등을 효율적으로 사용할 수 있도록 하나로 묶어놓은 일종의 패키지입니다!

Unit 01 | Intro

왜 딥러닝 프레임워크를 사용할까요?

이미 검증된 알고리즘을 사용할 때마다 구현하는 것은 비효율적입니다.

딥러닝 프레임워크는 개발자가 쉽고 빠르게 핵심 알고리즘만 개발할 수 있도록 도와줍니다.

Unit 01 | Intro

딥러닝 프레임워크의 장점 1. 복잡해 보이는 모델을 쉽게 쓸 수 있습니다.

- 해당 모델의 구조를 자세히는 몰라도 쓸 수는 있습니다.

```
from efficientnet_pytorch import EfficientNet  
model = EfficientNet.from_pretrained('efficientnet-b7')
```



Unit 01 | Intro

딥러닝 프레임워크의 장점2. Gradients 계산이 쉽습니다.

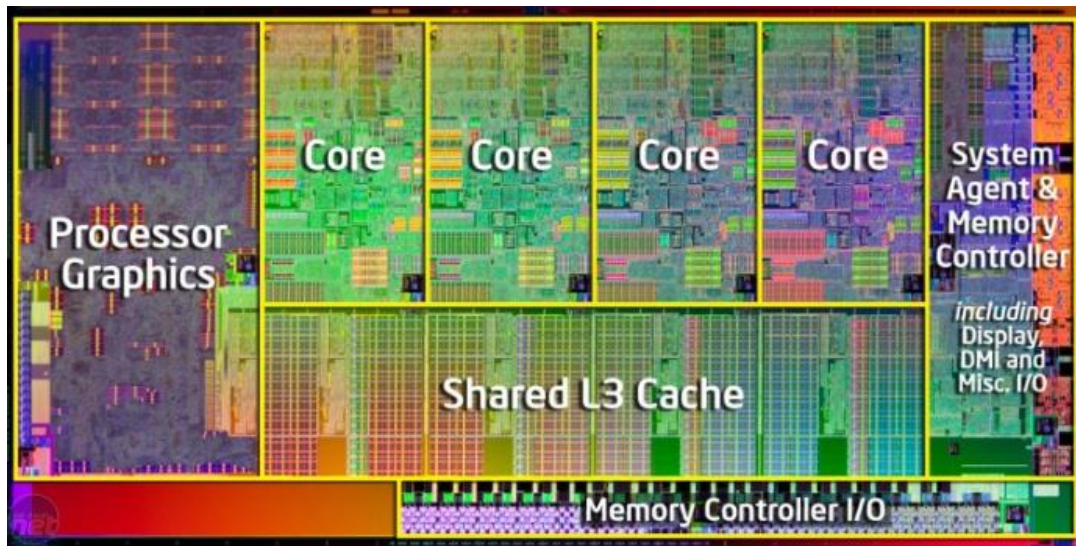
- Gradients를 자동으로 계산합니다.



Unit 01 | Intro

딥러닝 프레임워크의 장점3. GPU 활용이 쉽습니다.

- GPU를 사용하면 왜 좋을까요?



CPU는 코어가 16~32개



GPU는 코어가 매우 많다!

GPU의 코어는 단순하지만 코어가 많아서 한번에 많은 연산을 할 수 있다!

Unit 01 | Intro

막간의 팁! GPU 보는 법

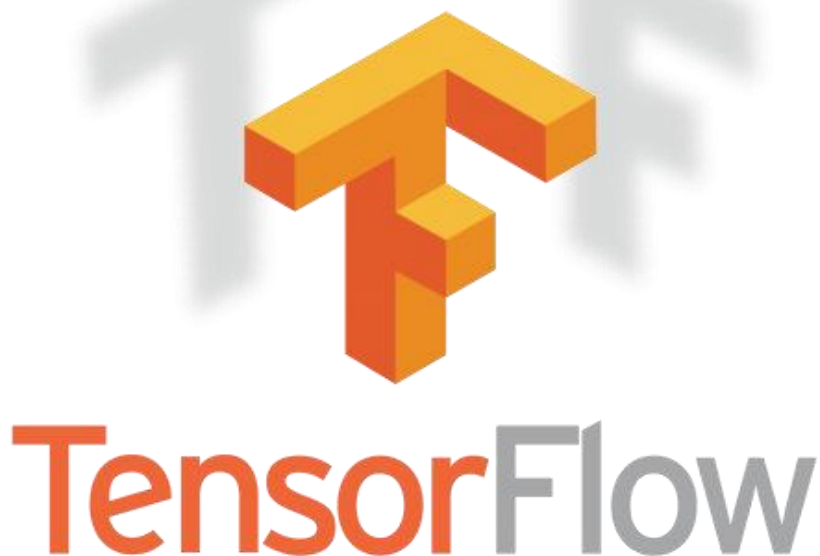
- Ti가 붙으면 프레임 버퍼가 11.0GB, 아니면 8.0GB인데 11.0GB면 더 큰 배치 사이즈를 메모리에 할당 가능합니다.

Year	Graphics Card	Price	Index	Average 1080p FPS	Average 1440p FPS	Average 4K FPS
2018	TITAN RTX	₩2,823,120.3	223	169.8	129.4	77.7
2018	RTX 2080 Ti	₩1,340,953.9	213	165.3	126	75.6
2019	RTX 2080 SUPER	₩789,660.3	184	150	113.2	67.6
2017	TITAN V	₩3,387,970.3	180	143.8	109.6	67.1
2018	RTX 2080	₩782,882.1	170	141.8	105.9	62.7
2019	RTX 2070 SUPER	₩563,720.3	156	132.1	97.5	58.1
2017	GTX 1080 Ti	₩911,667.9	156	132.9	101	60.3
2017	TITAN Xp	₩1,354,510.3	154	130.3	97.5	59.5
2019	Radeon VII	₩750,120.8	150	130.3	96.8	56.8
2019	RX 5700 XT	₩450,750.3	148	126.9	94.2	55.2
2018	RTX 2070	₩529,829.3	145	125.2	91.1	55

Unit 01 | Intro

프레임워크에는 뭐가 있나요?

- 첫번째 친구는 Tensorflow입니다!



- 이름 : 텐서플로우
- 구글 브레인팀에서 제작
- 데스크 탑이나 서버는 물론 안드로이드와 같은 모바일환경에서 강세를 보이고 있다. (Serving)
- Mutli-GPU Support가 잘 되어 있다.
- 최근 2.0으로 업데이트를 하면서 더 높은 수준의 API를 지원

Unit 01 | Intro

프레임워크에는 뭐가 있나요?

- 두번째 친구는 Pytorch입니다!

The PyTorch logo, featuring the word "PYTORCH" in a bold, black, sans-serif font. The letter "O" is replaced by a stylized orange flame icon with a small purple dot at its base.

- 이름 : 파이토치
- 페이스북에서 제작
- 이해와 디버깅이 쉽고 직관적이다.
- Numpy스럽지만 쉬운 코딩이 가능하다!
- 유연~하다
- 연구할 땐 파이토치를 더 선호한다는 설문결과가 있다.

Unit 01 | Intro

프레임워크에는 뭐가 있나요?

- 세번째 친구는 Keras입니다!

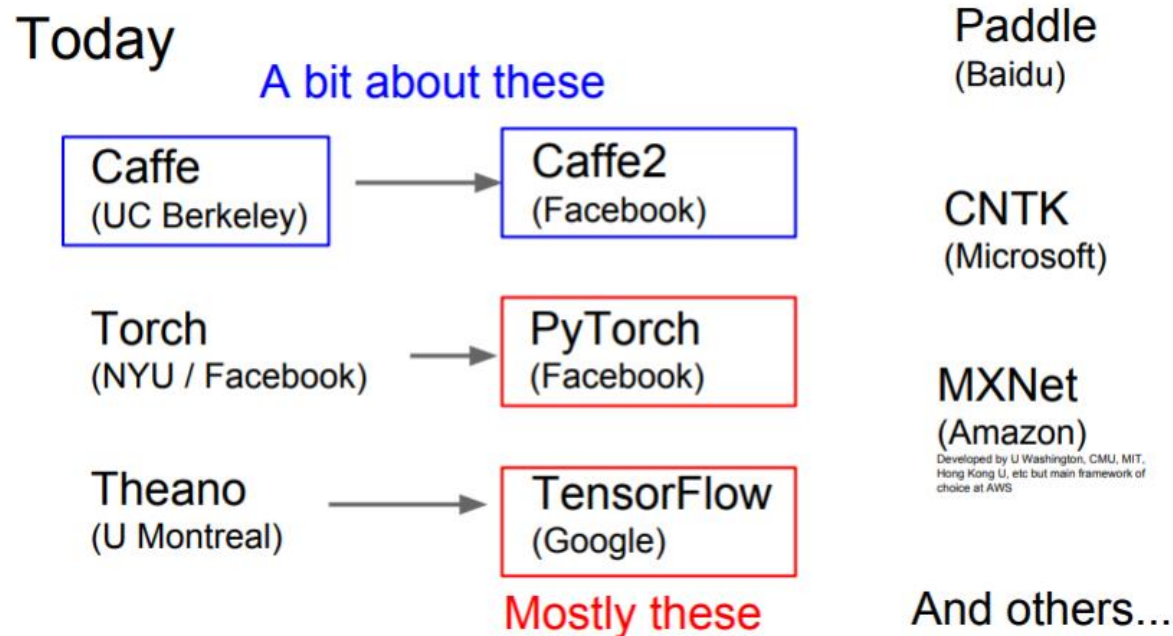


- 이름 : 케라스
- 구글이 제작한 건 아니지만 제작하고 나중에 합쳤다.
- 이해와 디버깅이 쉽고 직관적이다.
- TF의 높은 수준의 API가 요놈이다.
- ~~본인이 좋아합니다.~~

Unit 01 | Intro

프레임워크 관계도

- 이렇게 서로 친한 애들입니다!

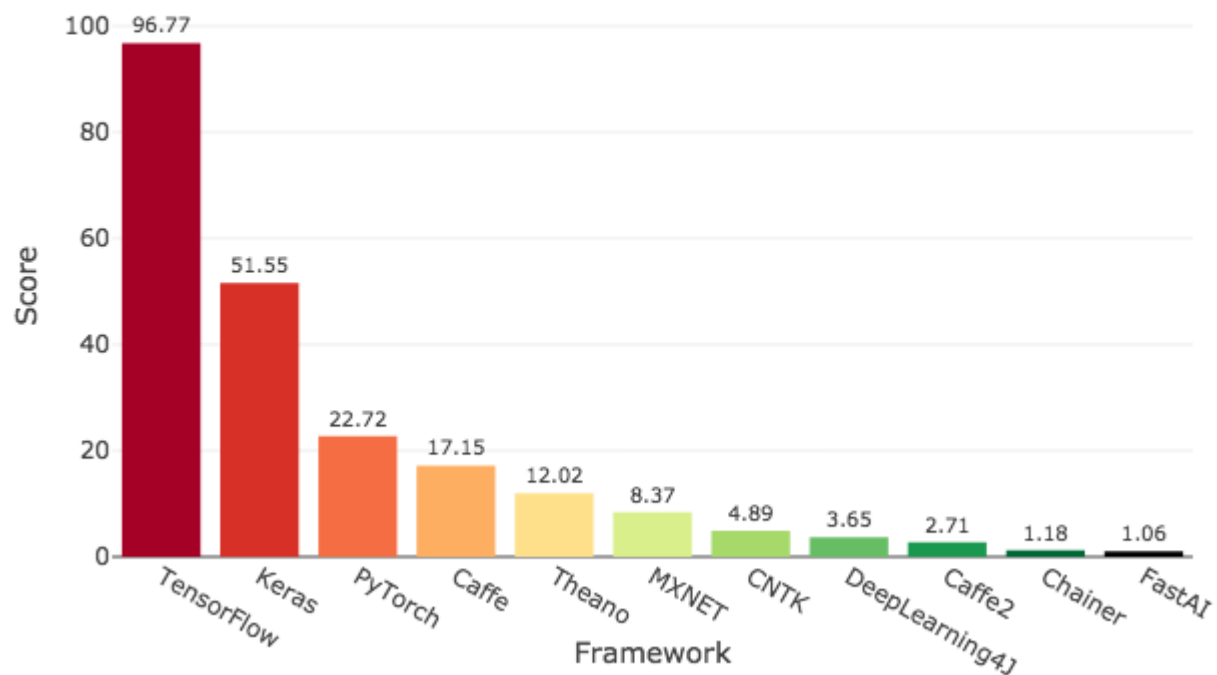


Unit 01 | Intro

프레임워크 선호도

- TF가 제일 인기가 많네요!

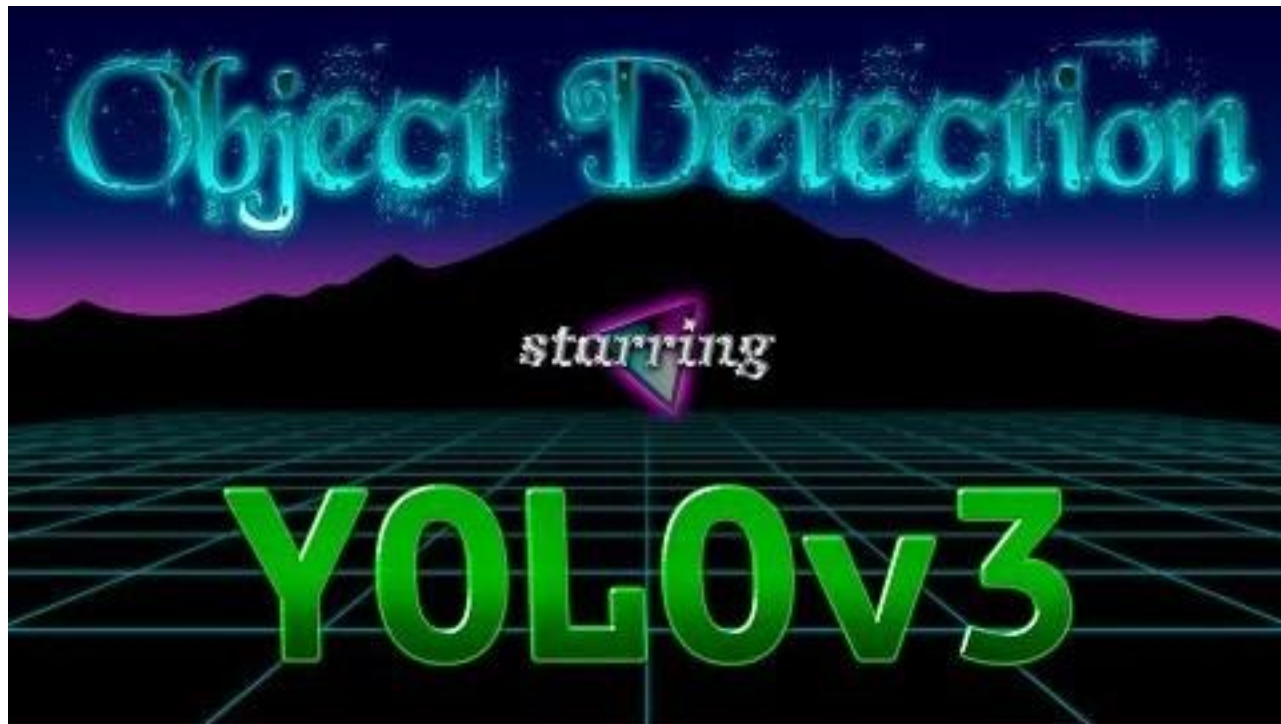
Deep Learning Framework Power Scores 2018



Unit 01 | Intro

주의 : 프레임워크에 너무 얽매이지 맙시다!

- 아래 모델은 빠른 추론을 위해 C언어로 만들어져 있습니다!



Unit 02 | Overview – Tensorflow VS Pytorch

Tensorflow 2.0 설치 시 주의할 점

- 로컬 설치 시 CUDA Version을 확인하는 것이 가장 중요합니다!

Software requirements

The following NVIDIA® software must be installed on your system:

- [NVIDIA® GPU drivers](#) [🔗](#) - CUDA 10.1 requires 418.x or higher.
- [CUDA® Toolkit](#) [🔗](#) - TensorFlow supports CUDA 10.1 (TensorFlow >= 2.1.0)
- [CUPTI](#) [🔗](#) ships with the CUDA Toolkit.
- [cuDNN SDK](#) [🔗](#) (>= 7.6)
- (Optional) [TensorRT 6.0](#) [🔗](#) to improve latency and throughput for inference on some models.

Unit 02 | Overview – Tensorflow VS Pytorch

Pytorch 설치 시 주의할 점

- 홈페이지에서 버전 확인을 잘 합니다!

QUICK START
LOCALLY

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, 1.5 builds that are generated nightly. Please ensure that you have **met the prerequisites below (e.g, numpy)**, depending on your package manager. Anaconda is our recommended package manager since it installs all dependencies. You can also [install previous versions of PyTorch](#). Note that LibTorch is only available for C++.

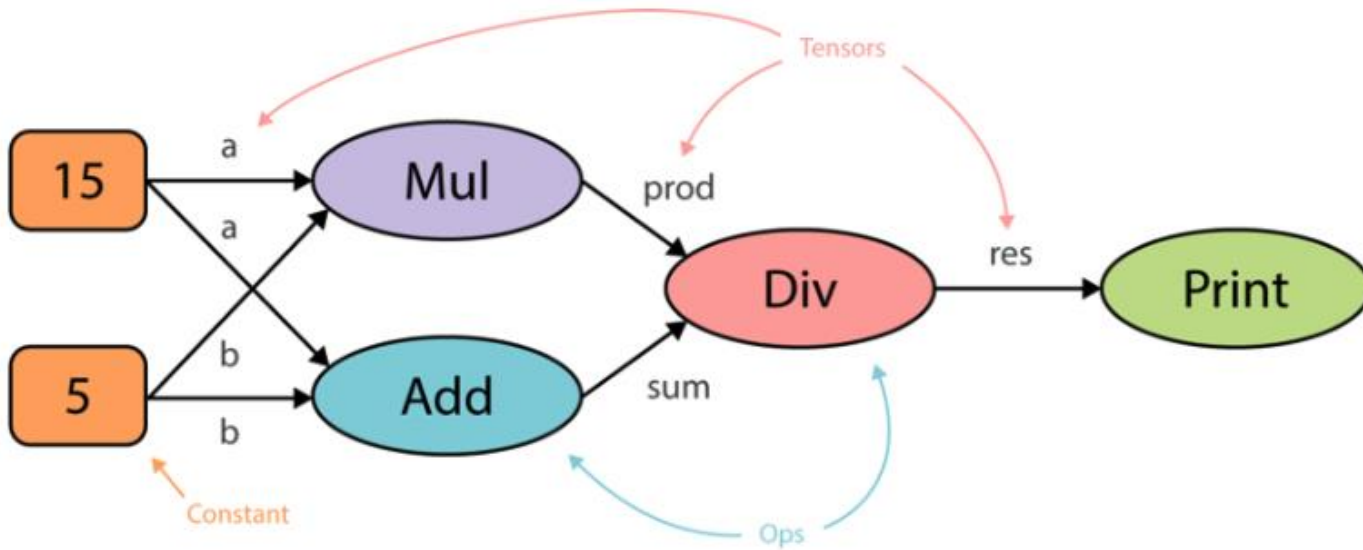
PyTorch Build	Stable (1.4)		Preview (Nightly)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python		C++ / Java	
CUDA	9.2	10.1	None	
Run this Command:	<code>conda install pytorch torchvision cudatoolkit=10.1 -c pytorch</code>			

[Previous versions of PyTorch](#)

Unit 02 | Overview – Tensorflow VS Pytorch

TF와 Pytorch의 메커니즘 차이 1 – Static vs Dynamic Graph Definition

- Tensorflow는 Static Graph 입니다!



그래프란?

정점과 간선으로 이루어져
있는 자료구조!

연산자와 피연산자가
옆 그림처럼 미리 그래프
형식으로 선언이 되고 Session을
통해 연산이 수행됩니다.

Session이란?

데이터마켓에서 세션만료
되신 적 있죠? 바로 그겁니다!
Session 동안 TF는 연산을 수행합니다.

Unit 02 | Overview – Tensorflow VS Pytorch

TF와 Pytorch의 메커니즘 차이 1 – Static vs Dynamic Graph Definition

- Pytorch는 Dynamic Graph 입니다!

A graph is created on the fly

```
from torch.autograd import Variable  
  
x = Variable(torch.randn(1, 10))  
prev_h = Variable(torch.randn(1, 20))  
W_h = Variable(torch.randn(20, 20))  
W_x = Variable(torch.randn(20, 10))
```



Session 없이 동작으로 바로 연산을 수행할 수 있습니다!

즉, 좀더 Python 스타일로 코딩이 가능합니다!

Unit 02 | Overview – Tensorflow VS Pytorch

TF와 Pytorch의 차이 2 – 분산처리

- Pytorch 에 있는 분산/병렬처리를 TF에서도 구현할 수 있지만 조~금 더 어렵습니다.

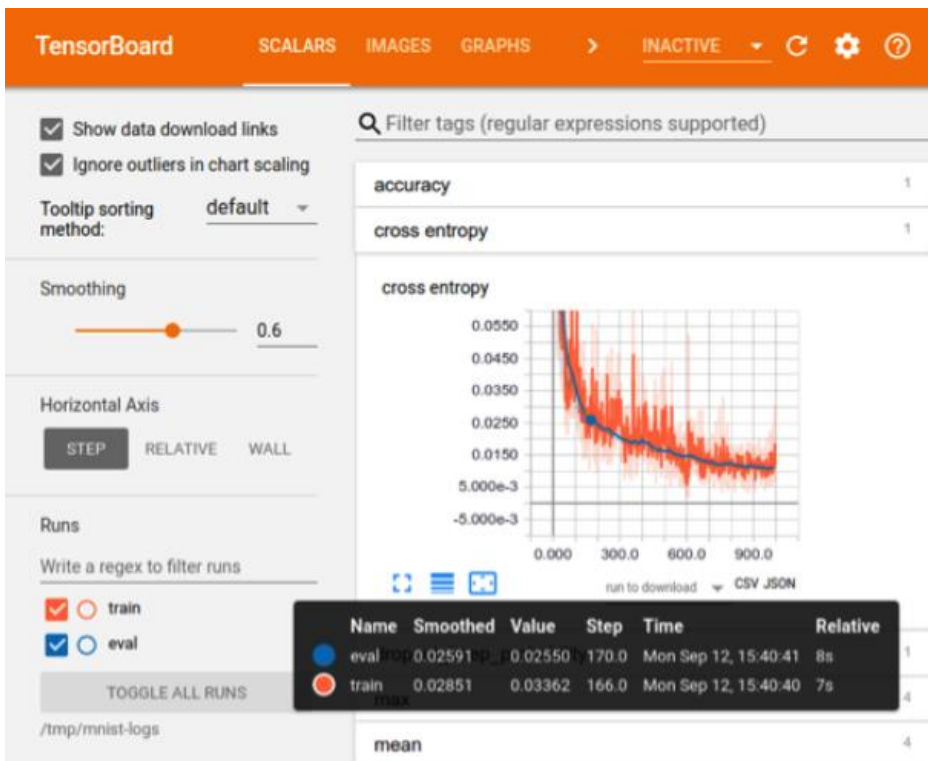
```
import torch.distributed as dist
from torch.nn.parallel import DistributedDataParallel

dist.init_process_group(backend='gloo')
model = DistributedDataParallel(model)
```

Unit 02 | Overview – Tensorflow VS Pytorch

TF와 Pytorch의 차이 3 – 시각화

- TF에 Tensorboard라는 것이 있어서 압승입니다.



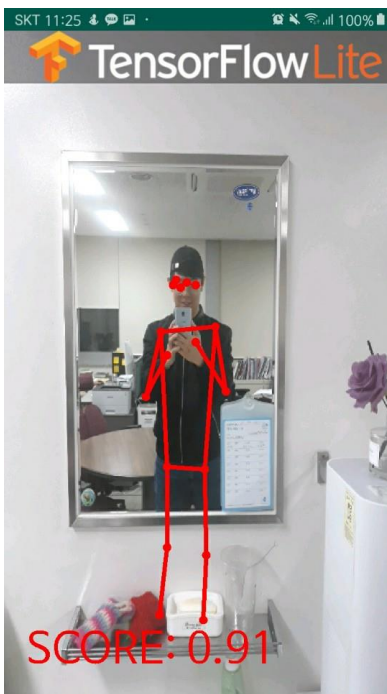
Why Good?

1. Loss 나 Metric을 쉽게 볼 수 있습니다.
2. 연산 그래프를 보기 쉽습니다.
3. 가중치들의 히스토그램들이 시간마다 변하는 걸 확인 가능합니다.
4. 이미지, 텍스트, 오디오 등을 보여줍니다.

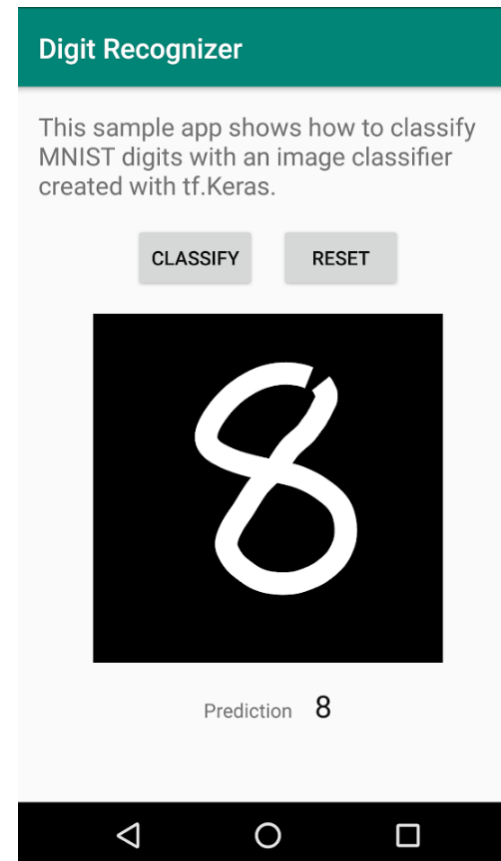
Unit 02 | Overview – Tensorflow VS Pytorch

TF와 Pytorch의 차이 4 – 상품화

- TF에 Rest Client API 라는 것이 있어서 압승입니다.



<https://github.com/tensorflow/examples/tree/master/lite/examples/posenet/android>



<https://medium.com/@margaretmz/e2e-tfkeras-tflite-android-273acde6588>

Unit 03 | Tensorflow Tutorial 실습

버전은? 2.0으로!

그럼 이제 Tensorflow Tutorial 에 빠져 봅시다! 그전에!

Unit 03 | Tensorflow Tutorial 실습

실습 목적은 처음 해보신 분들이라면
공식 문서를 보는 힘을 기르실 수 있고
잘 하시는 분들이시라면
공식 문서 튜토리얼을 보며 리마인드 할 수 있기 때문에 선정하였습니다!

Unit 03 | Tensorflow Tutorial 실습

그럼 이제 해봅시다!
구글에 tensorflow tutorial을 검색해 주세요!

Unit 04 | Pytorch Tutorial 실습

그럼 이제 Pytorch Tutorial 에 빠져 봅시다!

Unit 05 | Assignment

Assignment.(캐글 참가하기):

- <https://www.kaggle.com/t/f8a384082e2a491eb6af4bddb2b81513>
- NN 심화에서 5개 챕터 중 최소 3개 기법으로(챕터 중복 가능) Test 성능 평가
- 코드 제출은 Pytorch (1개), Keras (1개)로 총 2개 하시면 됩니다.
- 우수과제 선정 기준
 - Activation, Weight Initialization, Batch normalization, Optimizer, Regularization에 대한 **고민의 흔적**(+1)
 - 제출한 테스트의 **ACC가** 높을수록 **우수과제 선정 확률 증가** (성능 점수 + 1)
 - **다른 캐글 노트북 참고**에 대한 가산점 (Research 점수 +1)
- * 그 밖에 할 수 있는 **멋있는 기술** 우대(GAN이나 Few shot Learning 다 인정+2)

미니 캐글 우승자 혜택(1등):



+ 조건에 부합하는 우수과제자(2명): 기프티콘



Unit 05 | Assignment

1. Kaggle.json 다운받기(Account->Create New API Token)

API

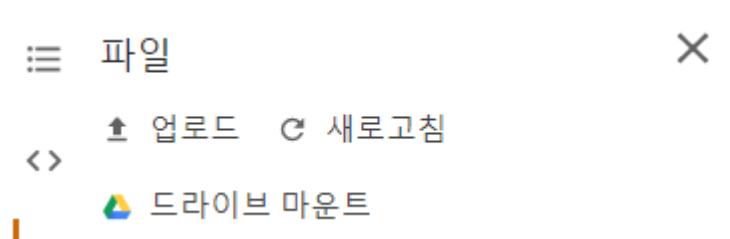
Using Kaggle's beta API, you can interact with Competitions and Datasets to download data, make submissions, and more via the command line. [Read the docs](#)

Create New API Token

Expire API Token

Unit 05 | Assignment

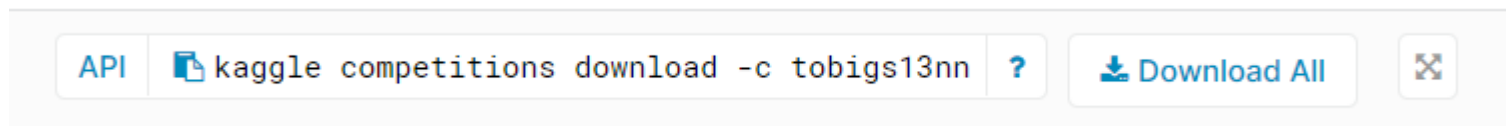
2. 자기 드라이브랑 연동하기



Unit 05 | Assignment

3. Colab Kaggle 연동하기

```
# 다운받은 API Token 업로드 하기
!pip install kaggle
from google.colab import files
files.upload()
# json 파일 옮겨주기
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
# Permission Warning 이 일어나지 않도록
!chmod 600 ~/.kaggle/kaggle.json
# json 파일 제대로 업로드 됐는지 확인
!ls -lha kaggle.json
# 본인이 참가한 모든 대회 보기
!kaggle competitions list
```



Q & A

들어주셔서 감사합니다.