

IITP AI- Challenge - 20200714

지난주 한일

1. 파이토치로 시작하는 딥러닝 기초 강의

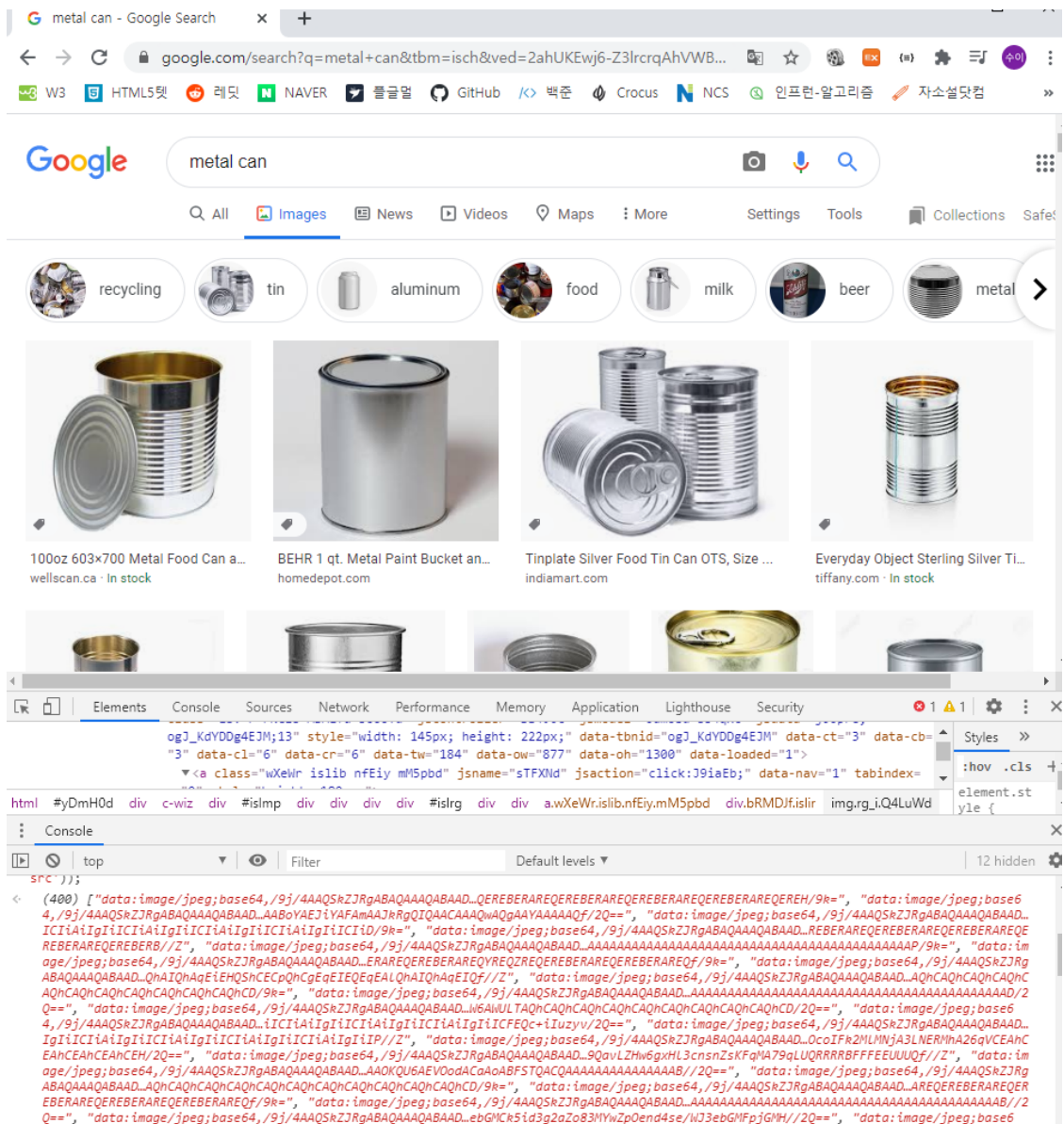
- Basic ML
 - pytorch 관련 함수 및 voca - view, squeeze, tensor, batch, stride, overfit, etc....
 - Classification
 - CE-softmax
 - BEC-sigmoid
 - tensor 조작시 같은 device 에 있어야지 계산 가능
 - CPU 에서 만든 tensor는 GPU에서 만든 tensor 와 연산불가능
 - multiple gpu의 경우에도 다른 gpu 에서는 서로 연산 불가능
- CNN
 - 진행 중..

2. fast.ai 강의

- chap 1 완료
- chap 2 완료
 - Creating your own dataset from Google Images
 - 이미지 수집 방법

```
urls=Array.from(document.querySelectorAll('.rg_i')).map(e1=>
e1.hasAttribute('src')?e1.getAttribute('src'):e1.getAttribute('data-
src'));
```

- step1 .구글 이미지 검색
- step2. 이미지 태그의 src or data-src 속성 값 추출
- step3. 사진 이미지 다운로드



3. 김소정 박사님 과제

TASK3 진행 완료.

TASK3:

기존에 공개된 waste-sorter 코드를 사용하여 모델을 학습하고 (CNN-pytorch)

F1 스코어 (recall 과 precision 의 조화평균. 찾아보면 계산법이 있습니다)

<https://towardsdatascience.com/how-to-build-an-image-classifier-for-waste-sorting-6d11d3c9c478>

실습문서 :

fastai.ipynb

<https://colab.research.google.com/drive/1ZPjj3Mvz1yImtpdMtaL2WDqA1bqii7#scrollTo=JONbSlrBckVF>

```
[23] learn.lr_find(start_lr=1e-6,end_lr=1e1)
learn.recorder.plot()
```

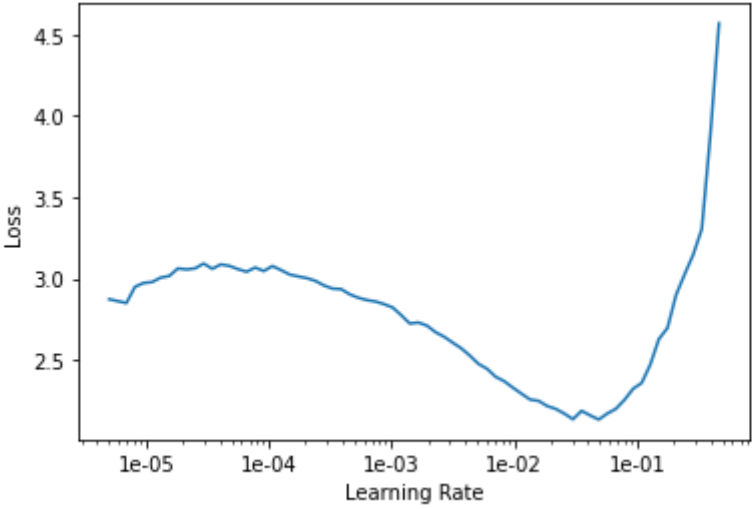
50.00% [1/2 00:36<00:36]

epoch train_loss valid_loss error_rate time

0	3.027514	#na#	00:36
---	----------	------	-------

10.26% [8/78 00:04<00:36 8.5154]

LR Finder is complete, type {learner_name}.recorder.plot() to see the graph.



learn.fit_one_cycle(20,max_lr=5.13e-03)

epoch	train_loss	valid_loss	error_rate	time
0	1.754059	0.683878	0.246032	00:42
1	1.088216	0.620009	0.200000	00:40
2	0.830537	0.654191	0.185714	00:40
3	0.871544	0.547270	0.171429	00:40
4	0.851565	0.672527	0.225397	00:40
5	0.780655	0.605888	0.182540	00:39
6	0.755507	0.768358	0.201587	00:40
7	0.711012	0.691535	0.180952	00:39
8	0.650654	0.490923	0.157143	00:40
9	0.589720	0.563562	0.174603	00:39
10	0.538456	0.399769	0.131746	00:40
11	0.489673	0.404601	0.133333	00:39
12	0.394011	0.328912	0.117460	00:39
13	0.347941	0.297039	0.096825	00:39
14	0.295323	0.272105	0.080952	00:39
15	0.238269	0.256565	0.082540	00:39
16	0.223883	0.253929	0.068254	00:39
17	0.188080	0.256358	0.073016	00:39
18	0.171349	0.249384	0.071429	00:40
19	0.157494	0.229591	0.061905	00:39

- Doing Wrong!!
 - valid_loss 가 높으면 learning rate(LR) 이 너무 높음 -> LR 을 낮춘다.
 - LR 이 너무 낮으면 error_rate 이 0.001 만큼 너무 작게 학습된다.
 - train_loss >> valid_loss : epoch 수가 적거나 LR 이 작거나
 - epoch 수가 너무 많은 경우 : error rate 이 getting worse 될때
- Doing Right!!
 - train_loss << valid_loss
 - error rate is improving
- 코드에 대한 리뷰..
 - LR 와 epoch 수를 잘 조작하면 원하는 결과를 얻을 수 있을 거 같다.
 - IITP 점수 방법으로 코드 - 아마존 플래닛찾아볼것

정확도 : Accuracy

정밀도 : Precision

재현율 : Recall

F1 Score : 정밀도와 재현율의 조화 평균

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

○

한마디로 표현해보면

정확도 : 예측이 정답과 얼마나 정확한가?

정밀도 : 예측한 것중에 정답의 비율은?

재현율 : 찾아야 할 것중에 실제로 찾은 비율은?

F1 Score : 정밀도와 재현율의 평균

다음주 할일

- 3강 - 멀티라벨, unet, darknet
- 3강 - 세그멘테이션 : 캔은 0, 유리 1 라벨링 된 데이터를 학습
700 개 정도의 데이터, 동영상 데이터인지 이미지 데이터인지 확인
- dataset training
- waste-sorter code 를 이용하면 되는지..
- 테스트 환경 업그레이드 필요!
 - 11 기가 GPU 메모리 - batch size 64
 - 8 기가 GPU 메모리 - batch size 32
 - colab -> colab Pro

<https://itkmj.blogspot.com/2020/03/colab-pro.html>

https://www.leadergpu.com/tensorflow_resnet50_benchmark

Item	Quantity	Price
Colab Pro	1	US\$9.99
Subscription monthly charge		

구독을 취소할 때까지 **매달 US\$9.99 및 관련 세금**이 청구됩니다. 체험 기간이 있는 구독의 경우 체험 기간이 끝나면 청구가 시작됩니다.

주 판매세(6.35%) US\$0.63

US\$10.62

결제 완료

Tesla P100-PCIE

지침을 따르니 아래와 같이 Tesla P100-PCIE가 감지 되었습니다. 500만원 이상 하는 고가의 GPU 장비입니다. 또는 T4 GPU로도 할당이 된다고 합니다.

일반 유저에게 지급되는 K80은 가격이 약 200만원 정도 합니다.

```
▶ gpu_info = !nvidia-smi
gpu_info = '\n'.join(gpu_info)
if gpu_info.find('failed') >= 0:
    print('Select the Runtime → "Change runtime type" menu to enable a GPU accelerator, ')
    print('and then re-execute this cell.')
else:
    print(gpu_info)
```

Tue Mar 3 12:08:38 2020

```
+-----+
| NVIDIA-SMI 440.48.02      Driver Version: 418.67      CUDA Version: 10.1      |
+-----+-----+
| GPU   Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+
|    0   Tesla P100-PCIE...    Off | 00000000:00:04.0 Off |             0         |
| N/A   46C    P0      29W / 250W |  0MiB / 16280MiB |      0%      Default |
+-----+-----+

+-----+
| Processes:                               GPU Memory |
|  GPU       PID    Type    Process name                     Usage      |
+-----+-----+
| No running processes found               |
+-----+
```

고용량 RAM

일반 RAM은 13.7기가 바이트로 감지되었습니다.

```
from psutil import virtual_memory
ram_gb = virtual_memory().total / 1e9
print('Your runtime has {:.1f} gigabytes of available RAM\n'.format(ram_gb))
```

Benchmark GitHub hash: 9165a70

Command: # python3.5 benchmarks/scripts/tf_cnn_benchmarks/tf_cnn_benchmarks.py --num_gpus=2 --model resnet50 --batch_size 32 (optional 64, 128, 256, 512)

Model: ResNet50

Date of testing: June 2017

The test results are shown in the following diagram:



질문!

- ResNet-50 으로 테스트를 할 것인지.

다음주 일정

- 화요일 4시 -
- 구글에서 다운받은 이미지 처리
- 소연님 trashnet 라벨링후 처리
- 코랩 프로
- 데이터 api - 내가원하는 포맷으로, 이미지 리사이징 - 연습을 많이 할것,
- annotation 하는 것으로