간단한 오픈소스(AlexNet) 분석하기

사실은 실행을 위한 튜토리얼+시각화

Presented by :
Hyeon Kang

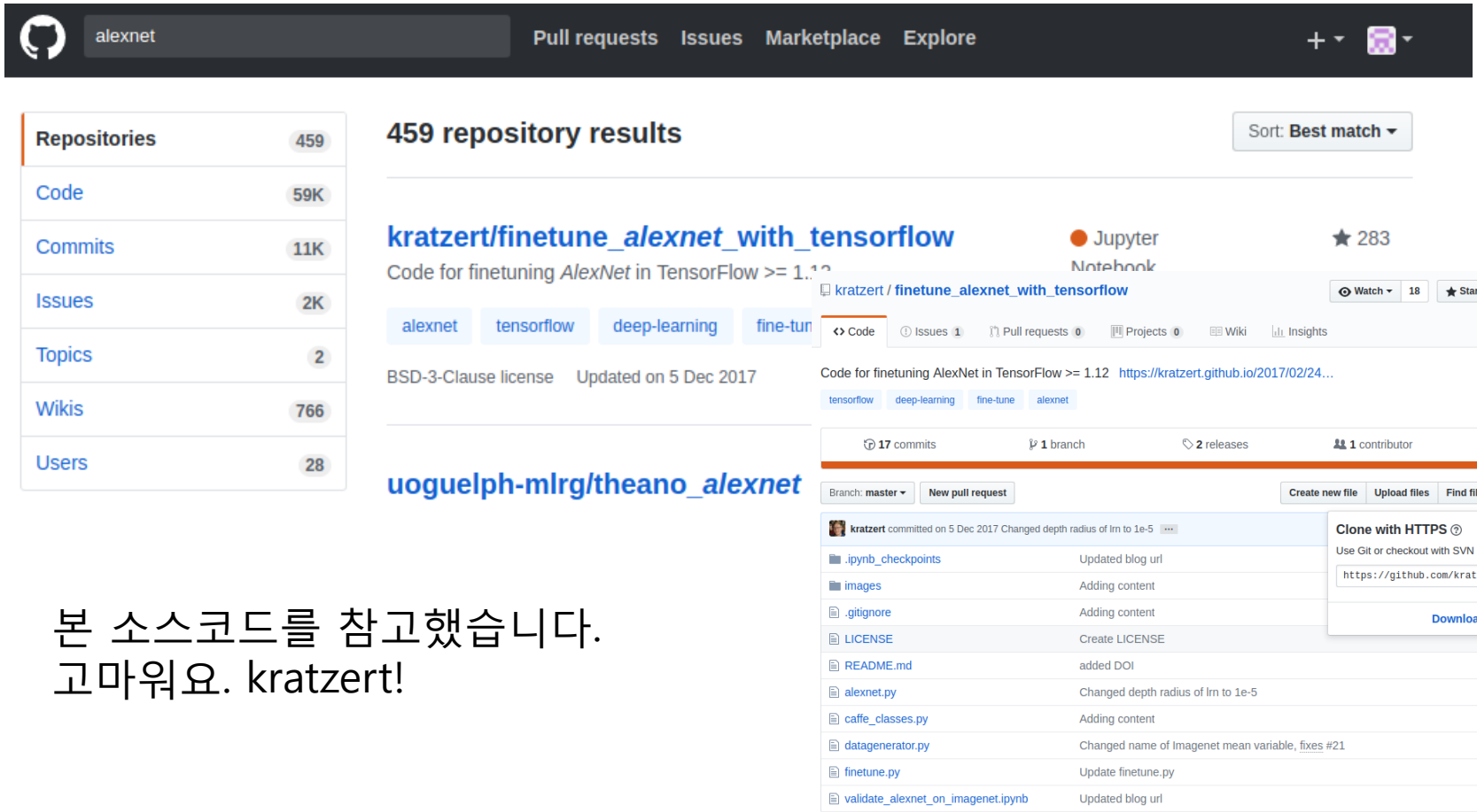
오픈소스(AlexNet) 분석하기

Github에는 오픈 소스가 많죠. 오픈 소스 분석이 공부에 대단히 도움이 됩니다.
오픈 소스 분석은 빌드부터 해야합니다.
비교적 간단한 소스코드라 아래 튜토리얼은 사실 아무것도 아니며 막막하실 때 한번 따라 하셔서 실행 먼저 해보시고 분석하시도록 만든 자료입니다.

라인 별로 로그 찍어가면서 학습하거나 모듈, 클래스, 메소드, 자료구조 및 알고리즘과 같이 Top-Down으로 학습하시면 TF 혹은 CNN의 구체적인 부분들이 명료해지리라 기대합니다.

문제가 생기면 issue 남겨주시거나 kang2000h@naver.com로 알려주시면 감사히 조치하겠습니다.

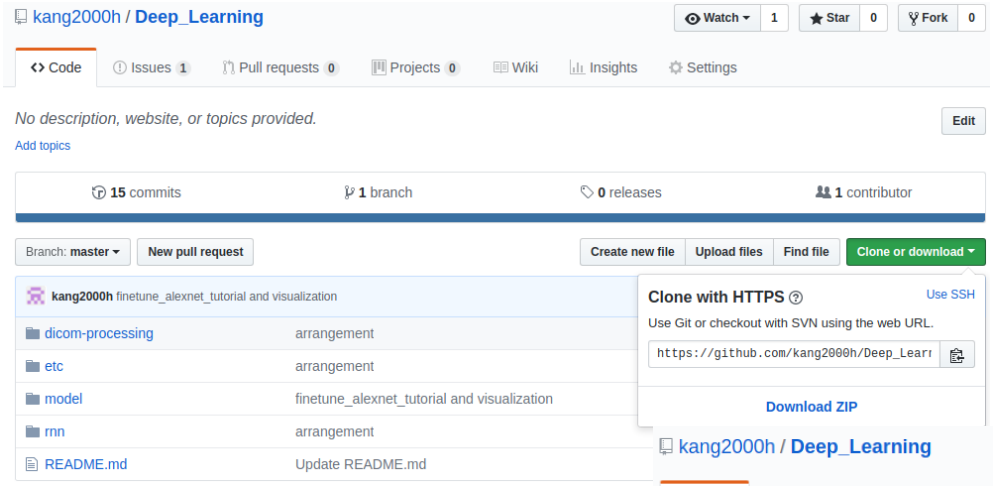
오픈소스(AlexNet) 분석하기



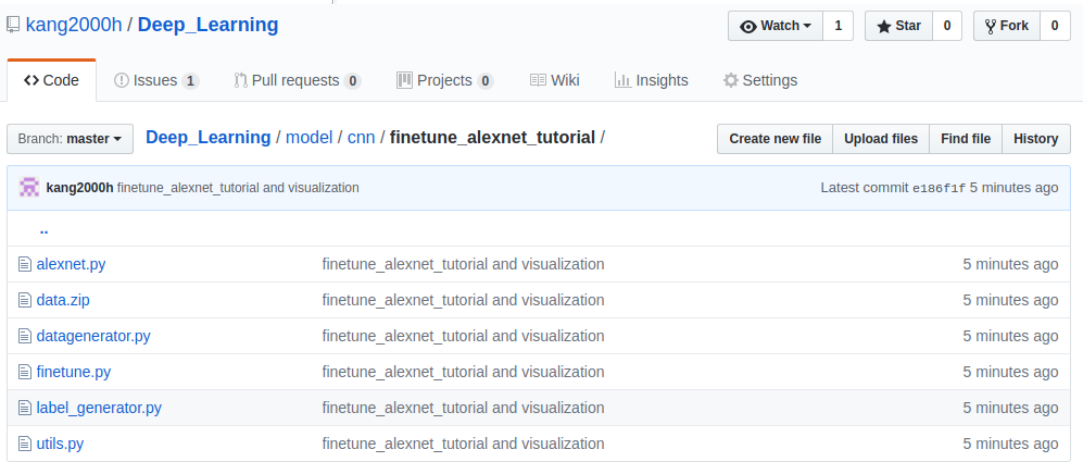
본 소스코드를 참고했습니다.
고마워요. kratzert!

https://github.com/kratzert/finetune_alexnet_with_tensorflow

오픈소스(AlexNet) 분석하기



갖고 놀기 위한 유틸을 조금 넣었습니다.
필요하시면 clone하시고
model/cnn/finetune_alexnet_tutorial
보시면 필요한 파일만 있습니다.
테스트용 영상(celebA)도 있고요.



https://github.com/kang2000h/Deep_Learning

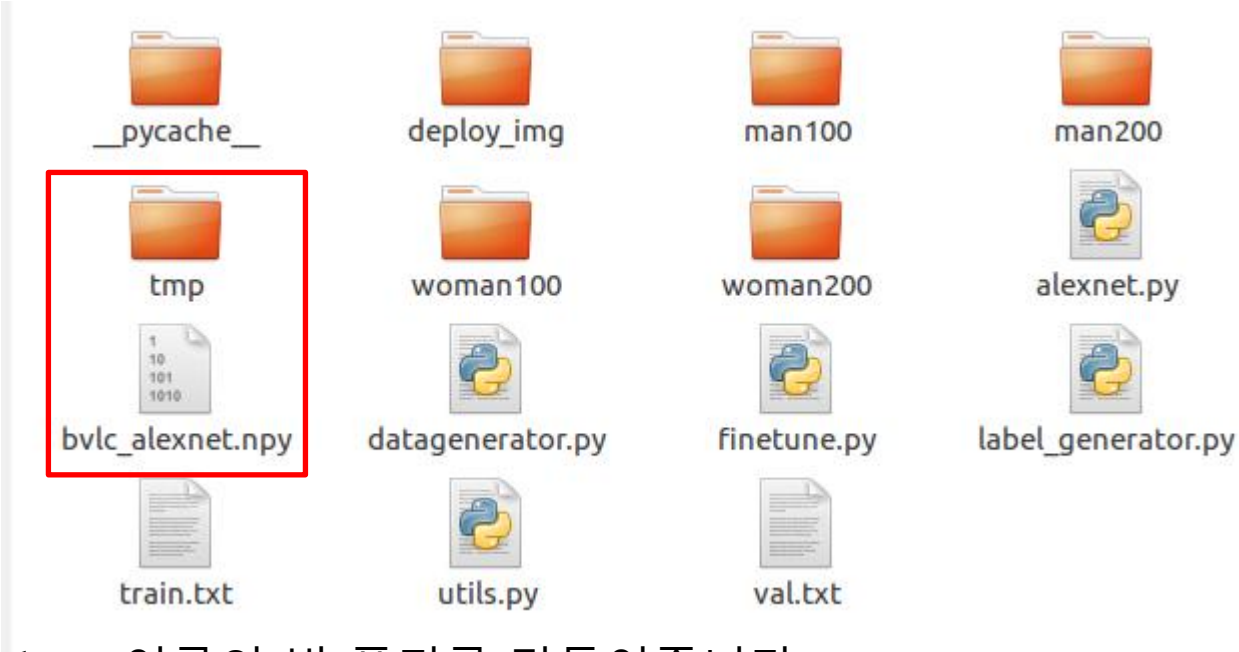
오픈소스(AlexNet) 분석하기

```
(tensorflow) root@donga-All-Series-Invalid-entry-length-16-Fixed-up-to-11:/home/
data/cnn_visualization# python label_generator.py --path_list ./man100 ./woman10
0 --train_rate 0.7
0 class num :  100
1 class num :  100
train_list 140
val_list 60
Do you want to save the paths?(Y/n)Y
Let me know save directory you want #^X
(tensorflow) root@donga-All-Series-Invalid-entry-length-16-Fixed-up-to-11:/home/
data/cnn_visualization# python label_generator.py --path_list ./man100 ./woman10
0 --train_rate 0.7
0 class num :  100
1 class num :  100
train_list 140
val_list 60
Do you want to save the paths?(Y/n)y
Let me know save directory you want to save #.
(tensorflow) root@donga-All-Series-Invalid-entry-length-16-Fixed-up-to-11:/home/
data/cnn_visualization#
```

python label_generator.py --path_list ./man100 ./woman100 --train_rate 0.7

위 명령을 사용해서 저장하겠냐고 하면 y, 어디에 저장하겠냐고 하면 .(;현재디렉토리)를 입력합니다. 그러면 각 데이터 폴더로부터 훈련/검증 셋을 7:3으로 나누어 train.txt/val.txt 파일을 만들어줍니다.

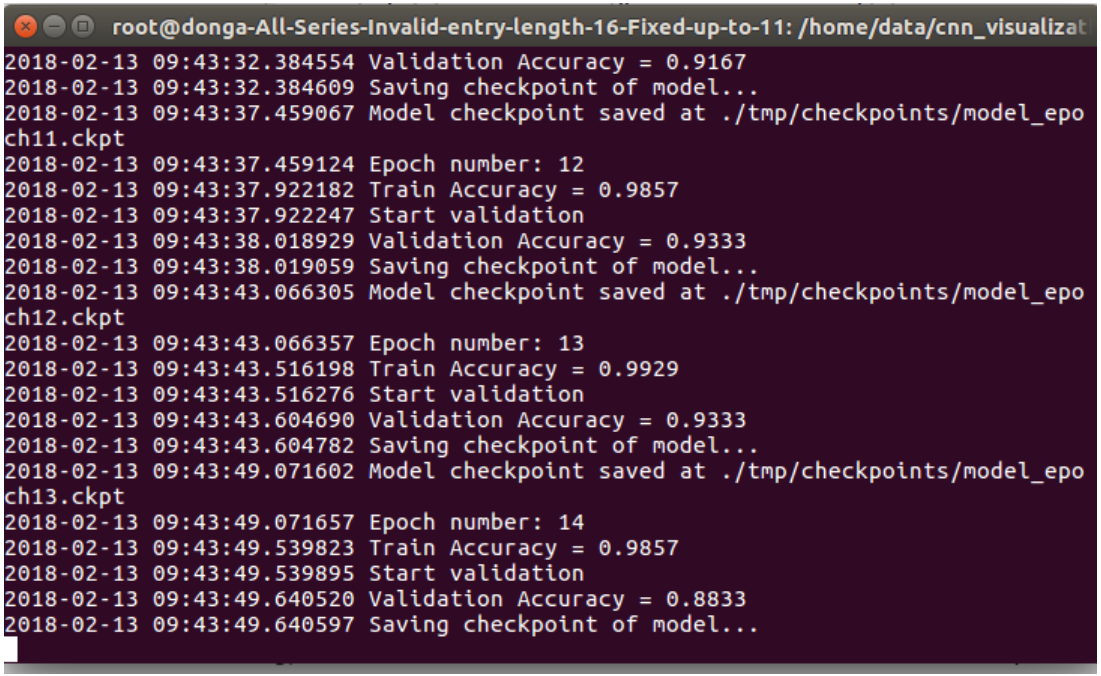
오픈소스(AlexNet) 분석하기



tmp 이름의 빈 폴더를 만들어줍니다. (소스 수정하셔서 자동으로 되게 하셔도 되죠. 저는 생략했습니다..)
data.zip 에서 데이터 폴더(man100, deploy_img etc)와 토론토 대학에서 제공하는 파라미터 파일(*.npy) 넣어줍니다.

http://www.cs.toronto.edu/~guerzhoy/tf_alexnet/

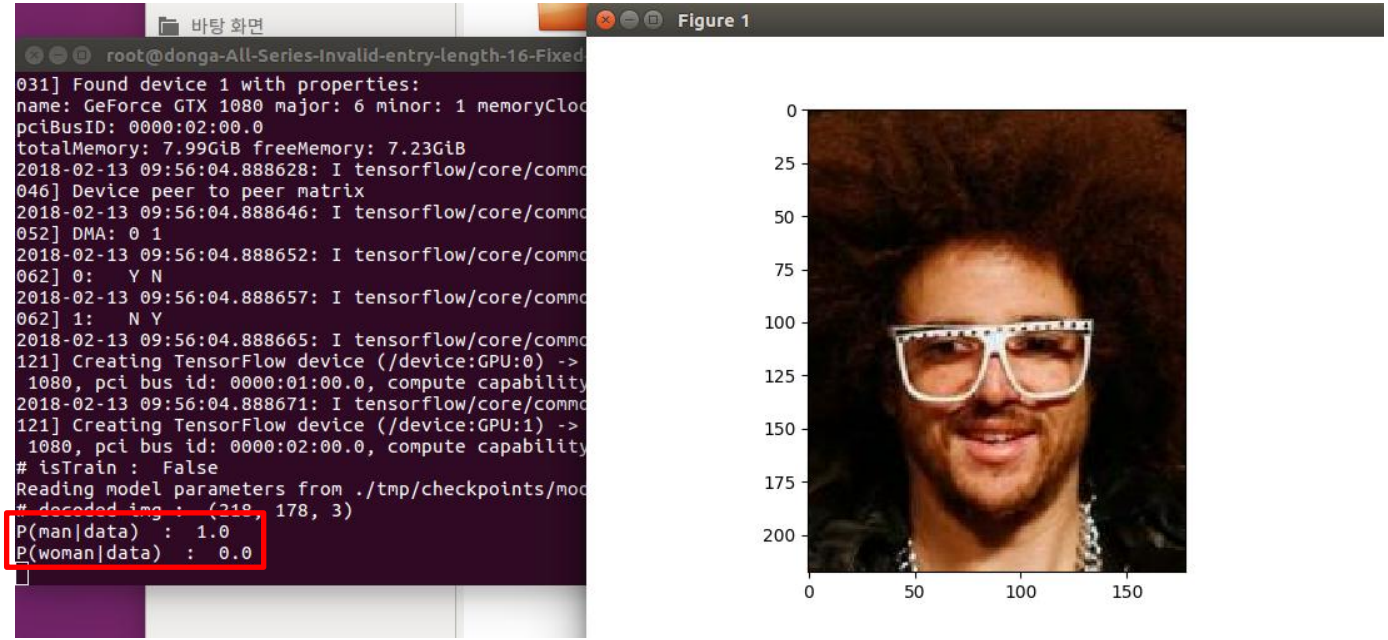
오픈소스(AlexNet) 분석하기



python finetune.py --istrain True

위 명령으로 훈련을 시켜줍니다. 검증단계 정확도가 90%에 근접하면 Ctrl +X 누르셔서 멈추시면 됩니다. tmp/checkpoints 디렉토리에 모델이 저장되었습니다.

오픈소스(AlexNet) 분석하기



python finetune.py --input_path ./deploy_img/m1.jpg -- istrain False

위 명령으로 한 장씩 테스트 해보겠습니다. 입력 영상과 함께 사후 확률을 출력합니다. 이미지를 바꿔가면서 출력해보세요. 다른 크기를 넣어도 되지만 잘 틀릴 수도 있습니다.

오픈소스(AlexNet) 분석하기

```
031] Found device 1 with properties:
name: GeForce GTX 1080 major: 6 minor: 1 memoryClockRate(GHz): 1.8
pciBusID: 0000:02:00.0
totalMemory: 7.99GiB freeMemory: 7.28GiB
2018-02-13 10:01:21.088044: I tensorflow/core/common_runtime/gpu/gp
046] Device peer to peer matrix
2018-02-13 10:01:21.088057: I tensorflow/core/common_runtime/gpu/gp
052] DMA: 0 1
2018-02-13 10:01:21.088063: I tensorflow/core/common_runtime/gpu/gp
062] 0:   Y N
2018-02-13 10:01:21.088068: I tensorflow/core/common_runtime/gpu/gp
062] 1:   N Y
2018-02-13 10:01:21.088077: I tensorflow/core/common_runtime/gpu/gp
121] Creating TensorFlow device (/device:GPU:0) -> (device: 0, nam
 1080, pci bus id: 0000:01:00.0, compute capability: 6.1)
2018-02-13 10:01:21.088083: I tensorflow/core/common_runtime/gpu/gp
121] Creating TensorFlow device (/device:GPU:1) -> (device: 1, nam
 1080, pci bus id: 0000:02:00.0, compute capability: 6.1)
# isTrain :  False
Reading model parameters from ./tmp/checkpoints/model_epoch21.ckpt
# decoded img :  (315, 242, 3)
P(man|data)  :  1.0
P(woman|data)  :  0.0
```

python finetune.py --input_path ./deploy_img/w10.jpg --visualize True

위 명령으로 영상을 바꿔가며 시각화 할 수 있습니다.
흑백으로 보고 싶으시면 소스코드 상에서 바꾸면 되는데 그냥 밝은 것은 흰색 남색은 검정색이라고 보시면 됩니다. imshow()에서 cmap='gray'추가하시면 됩니다.