CMU16824 Assignment1:
Object Classification with TensorFlow

Qian Gong(qgong1)

To Run code:
python xx.py VOCdevkit/VOC2007/

TASK 0: MNIST 10-digit classification in TensorFlow

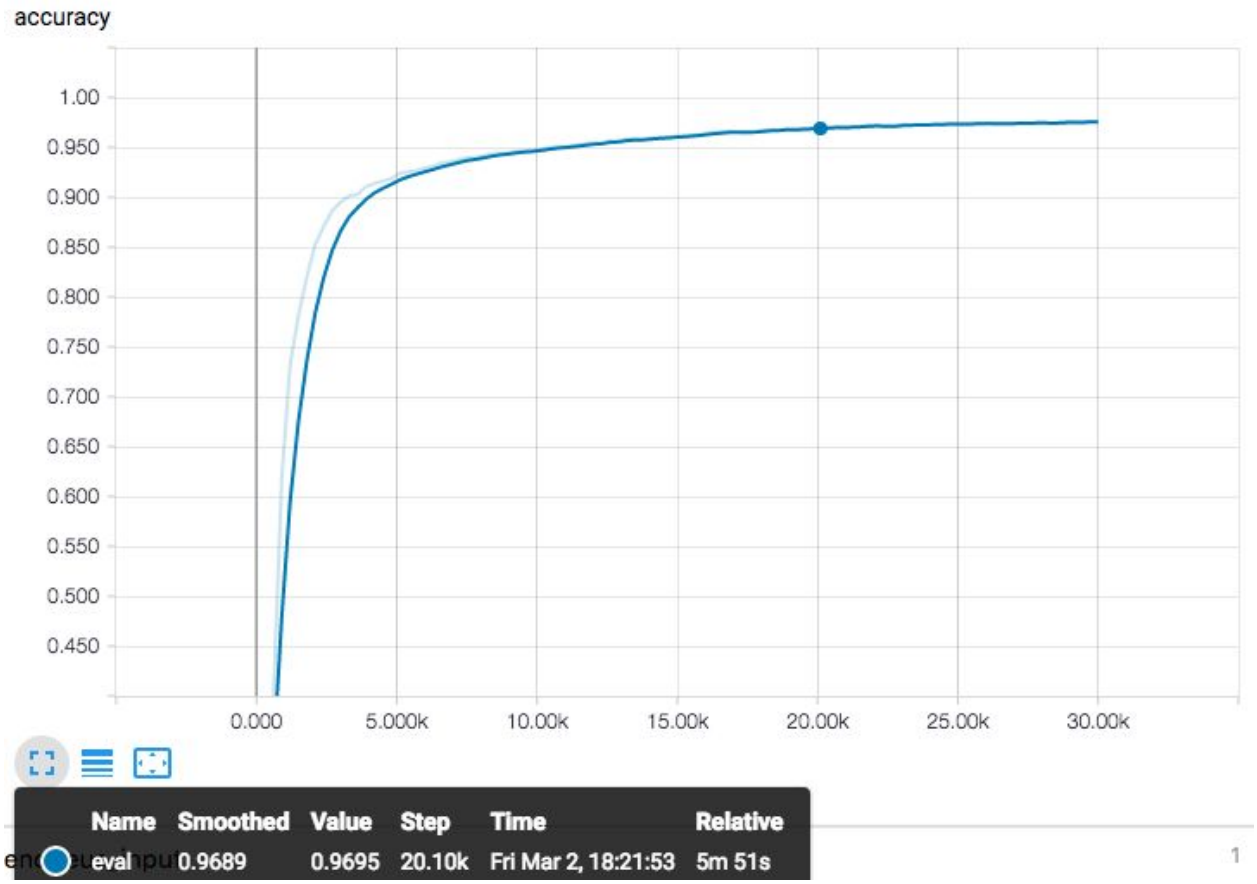
Q0.1:

```
INFO:tensorflow:Restoring parameters from ./tmp/mnist_convnet_model/model.ckpt-201
00
INFO:tensorflow:Finished evaluation at 2018-03-02-23:21:53
INFO:tensorflow:Saving dict for global step 20100: accuracy = 0.9695, global_step
= 20100, loss = 0.10538825
{'loss': 0.10538825, 'global_step': 20100, 'accuracy': 0.9695}
```

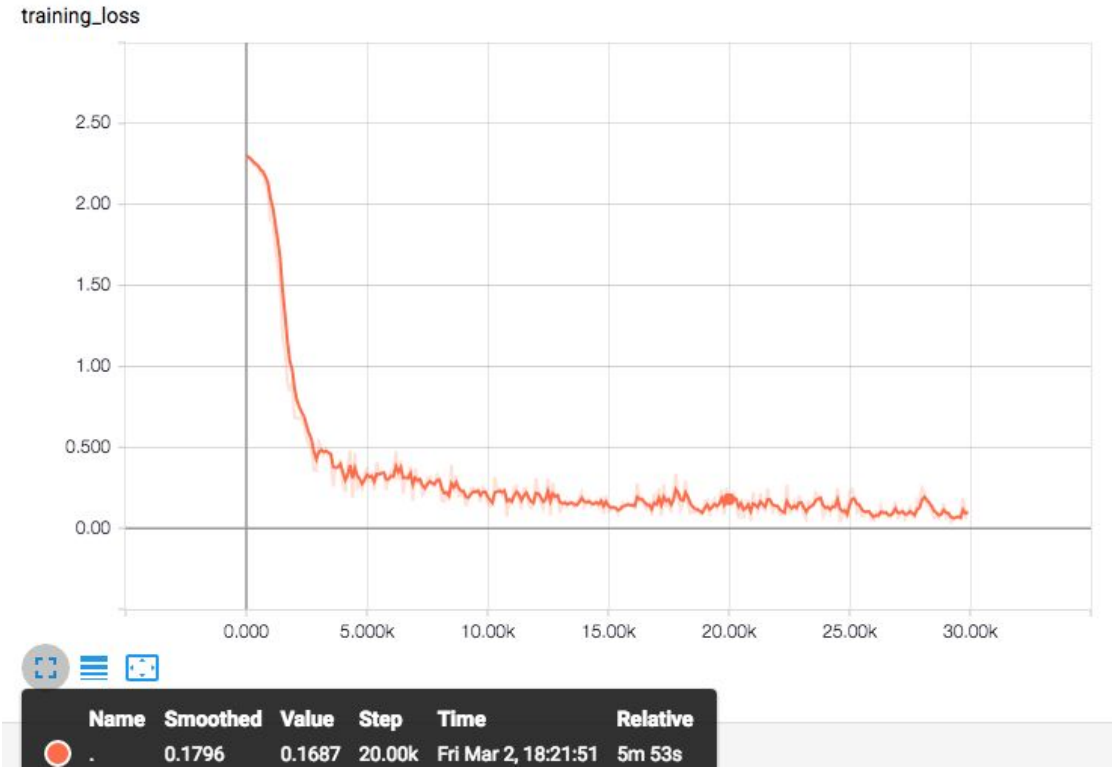
The test accuracy is 96.95%

Q0.2: From the figure we could find that, If train for more than 20000 iterations(eg. 30000), the train loss and accuracy won't improve much.

Q0.3:



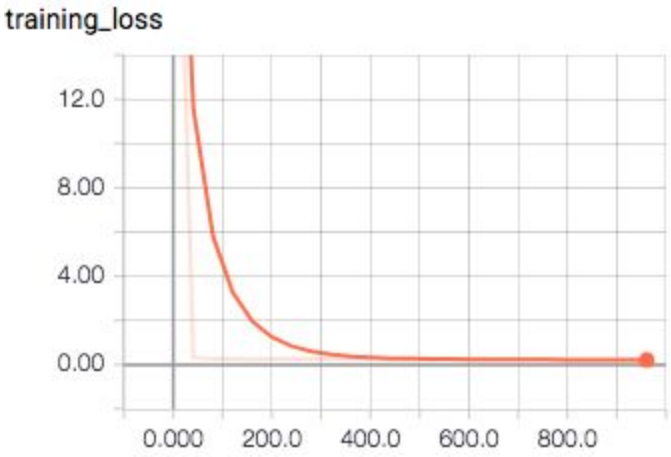
Test Accuracy at 20000 iter: **96.95%**



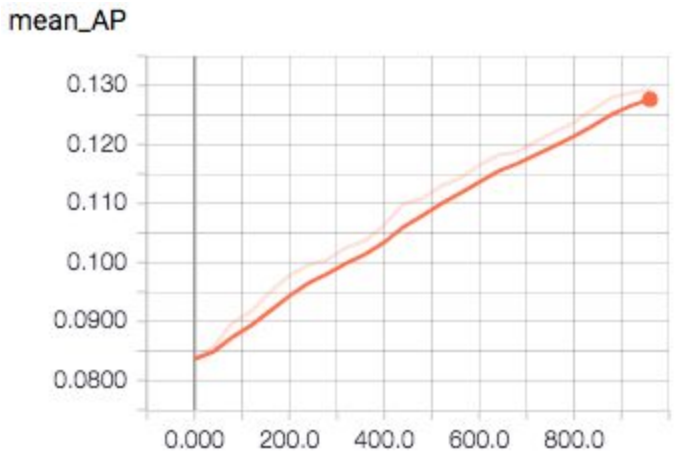
Train loss at 20000 iter: **0.1687**

TASK 1: 2-layer network for PASCAL multi-label classification

Train for 1000 iterations:



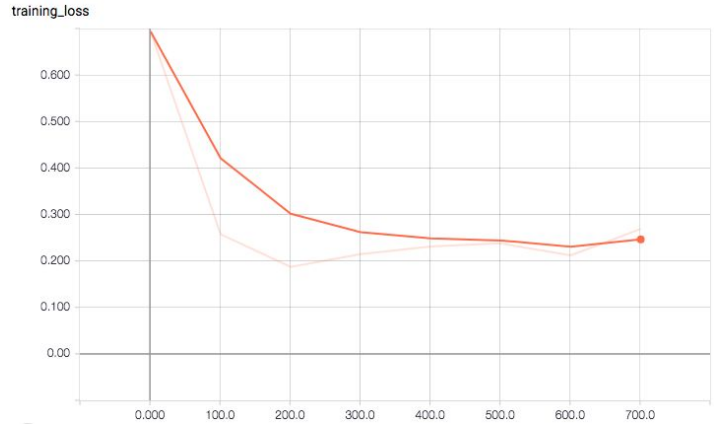
train loss at 1000 iter: **0.2166**



test mAP at 1000 iter: **0.1293**

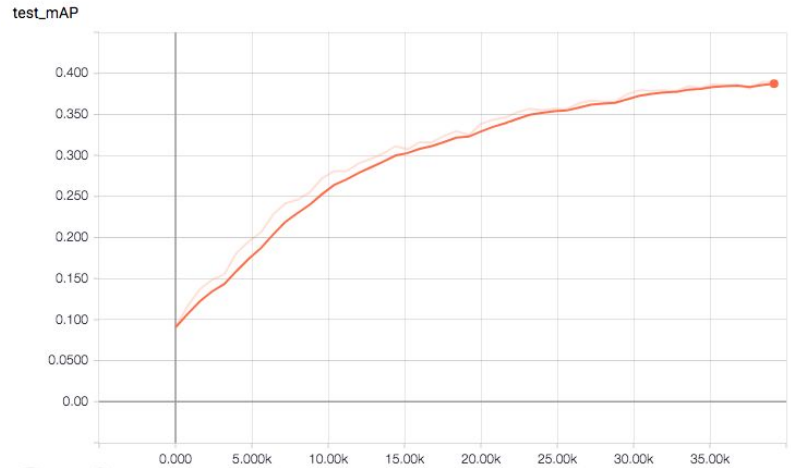
TASK2: AlexNet for PASCAL classification

Train for 40k iterations.



train loss at 40k iterations: **0.2462**

mean_AP:



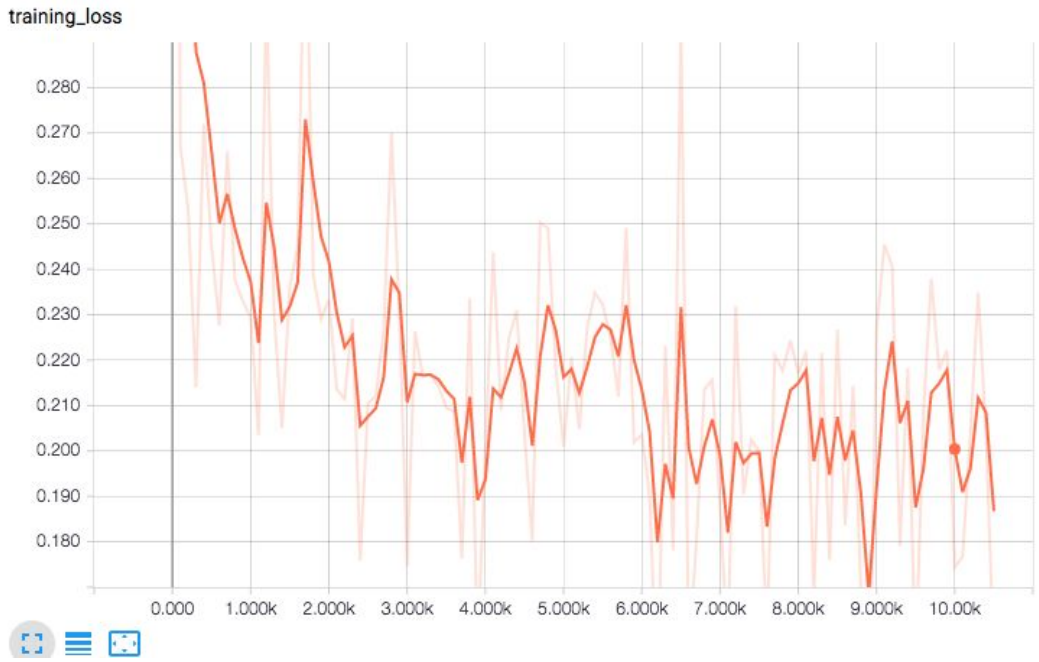
Test mAP at 40k iter: **0.3898**

Final Results for all classes:

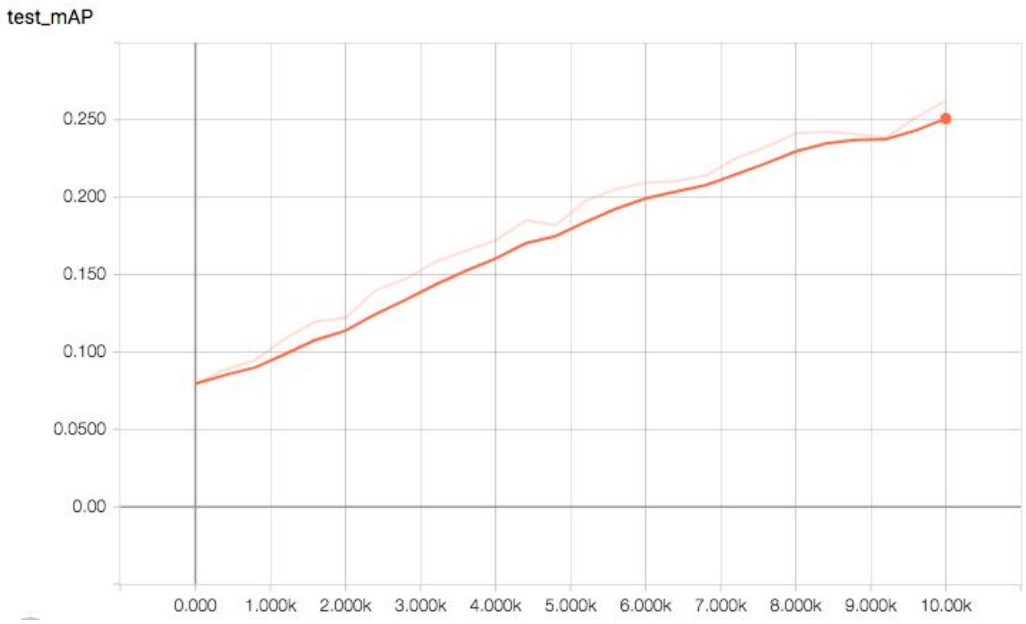
```
INFO:tensorflow:Restoring parameters from ./tmp/alexnet_model_scratch/model.ckpt-40000
Random AP: 0.0717540125144 mAP
GT AP: 1.0 mAP
Obtained 0.389810946052 mAP
per class:
aeroplane: 0.628607365414
bicycle: 0.368261498295
bird: 0.305081412739
boat: 0.381517785994
bottle: 0.176074881784
bus: 0.315480714905
car: 0.630637448458
cat: 0.341581183356
chair: 0.386343494802
cow: 0.169462387902
diningtable: 0.300630795769
dog: 0.285770217741
horse: 0.631297606899
motorbike: 0.503486353279
person: 0.788262219911
pottedplant: 0.179006651184
sheep: 0.276749044233
sofa: 0.314952510452
train: 0.494015763811
tvmonitor: 0.318999584103
```

TASK3: VGG-16 for PASCAL

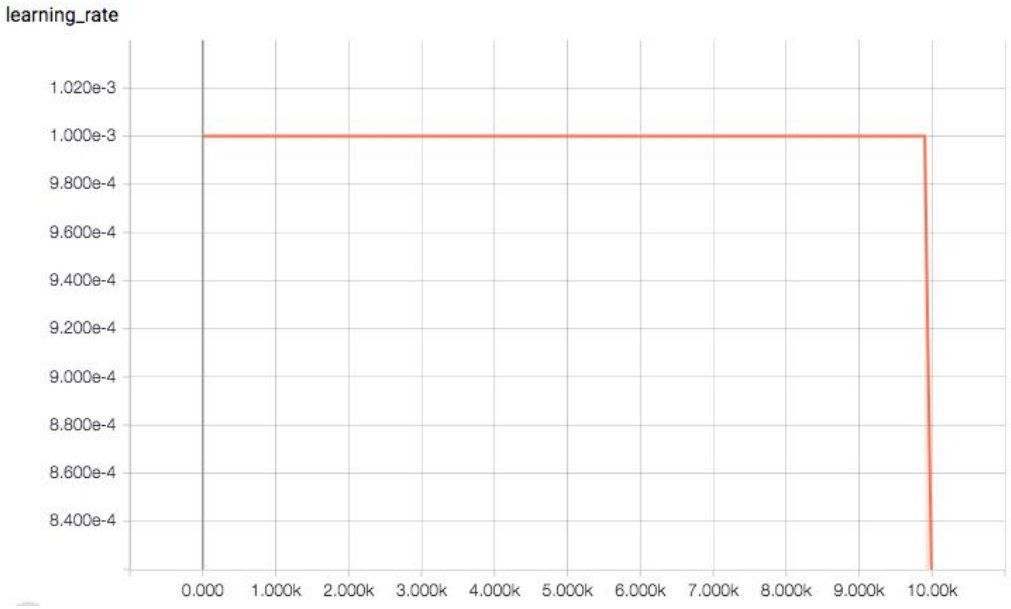
Count first 2 hours and trained 10k iterations



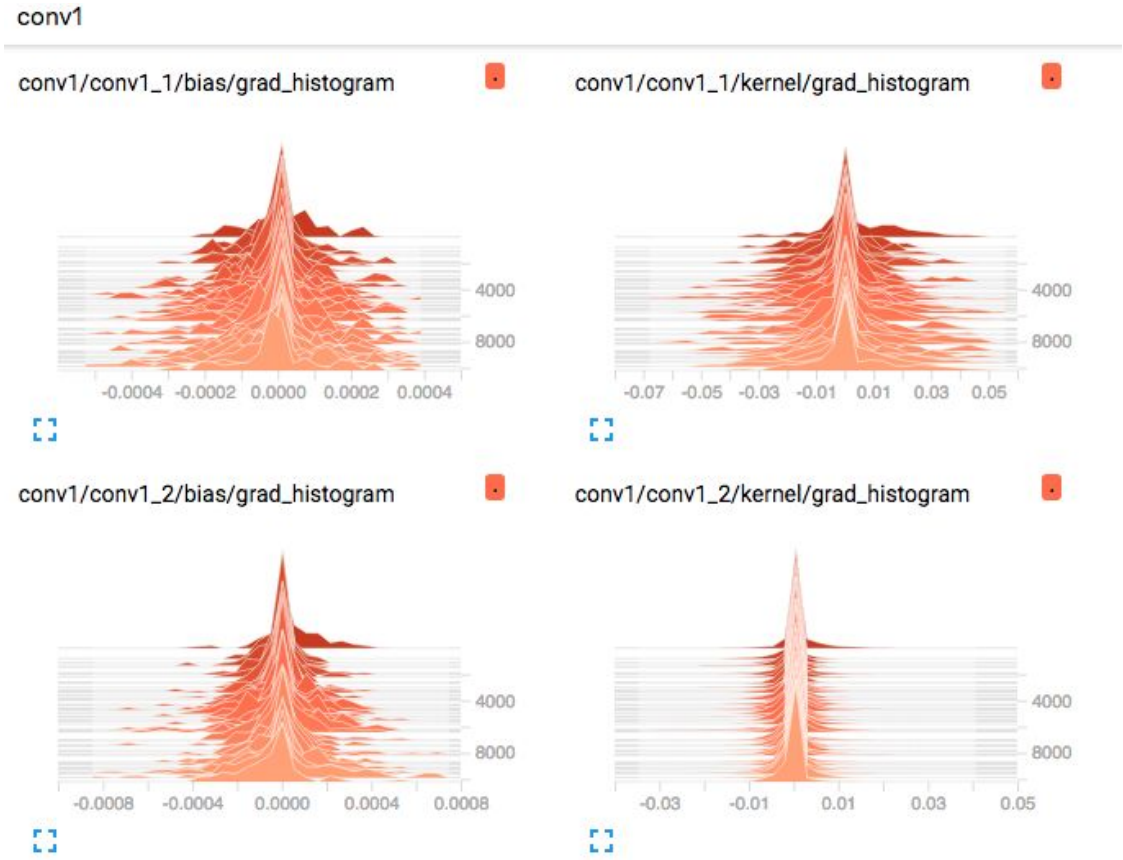
Train loss at 10k iteration: **0.1743**



Test mAP at 10k iteration: **0.2509**

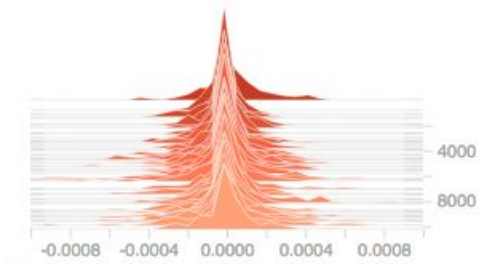


Histogram of gradients:

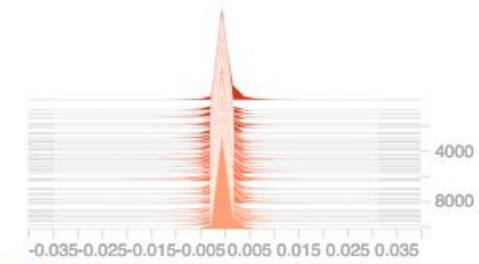


conv2

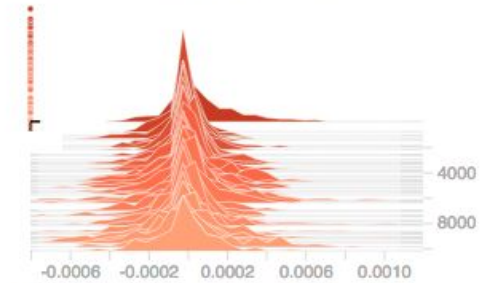
conv2/conv2_1/bias/grad_histogram



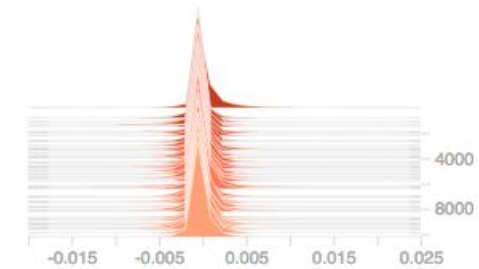
conv2/conv2_1/kernel/grad_histogram



conv2/conv2_2/bias/grad_histogram

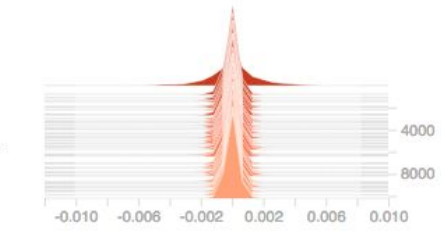


conv2/conv2_2/kernel/grad_histogram

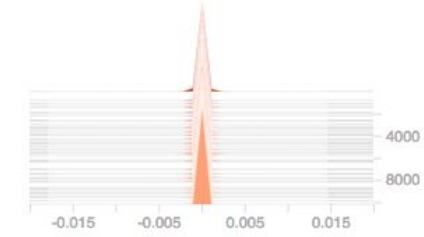


fc7

fc7/bias/grad_histogram

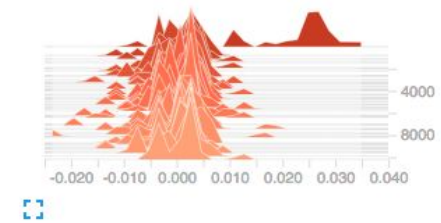


fc7/kernel/grad_histogram

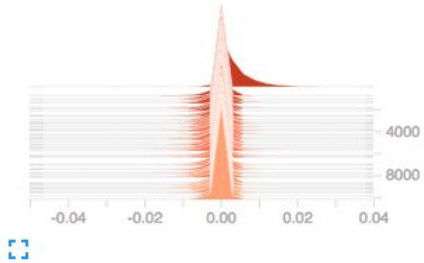


fc8

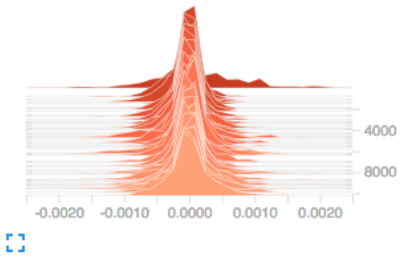
fc8/bias/grad_histogram



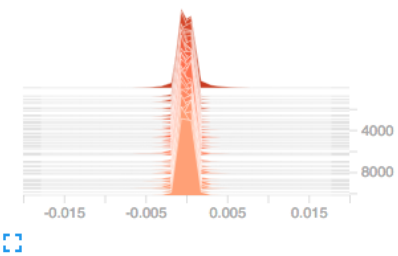
fc8/kernel/grad_histogram



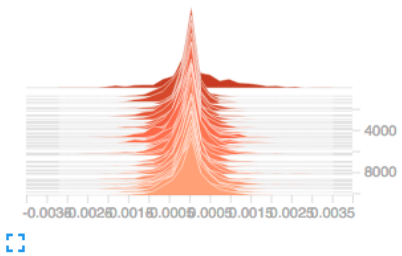
conv4/conv4_1/bias/grad_histogram



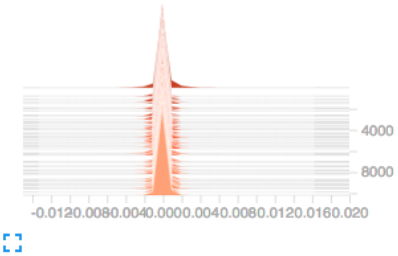
conv4/conv4_1/kernel/grad_histogram



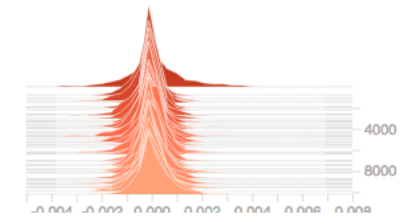
conv4/conv4_2/bias/grad_histogram



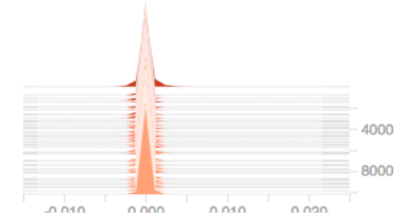
conv4/conv4_2/kernel/grad_histogram



conv4/conv4_3/bias/grad_histogram

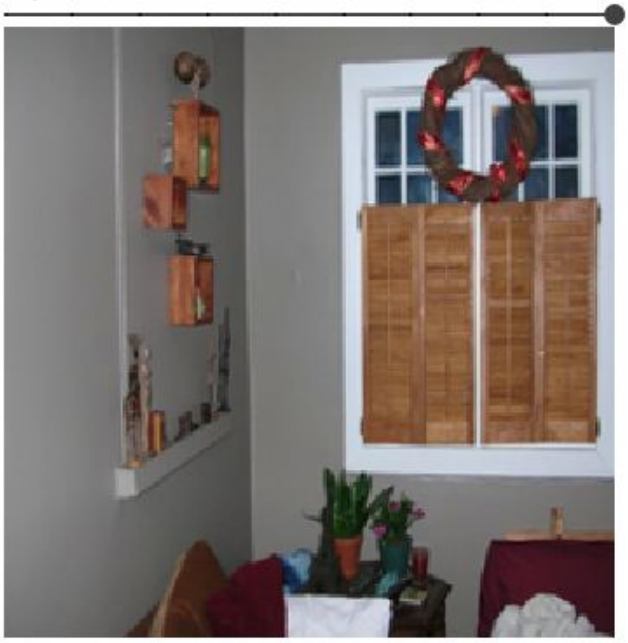


conv4/conv4_3/kernel/grad_histogram

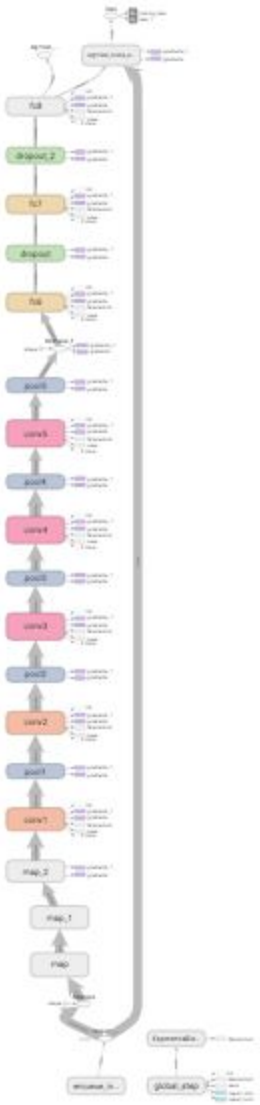


training_images

training_images/image/0
step **10,501** Thu Mar 01 2018 04:58:01 EST



training_images/image/1
step **10,501** Thu Mar 01 2018 04:58:01 EST

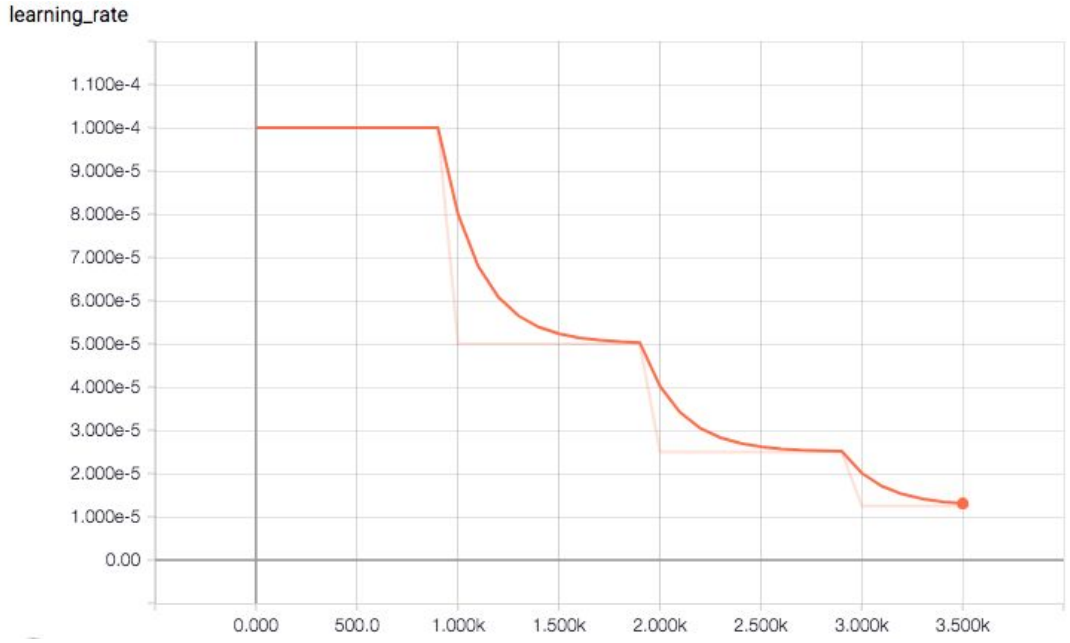


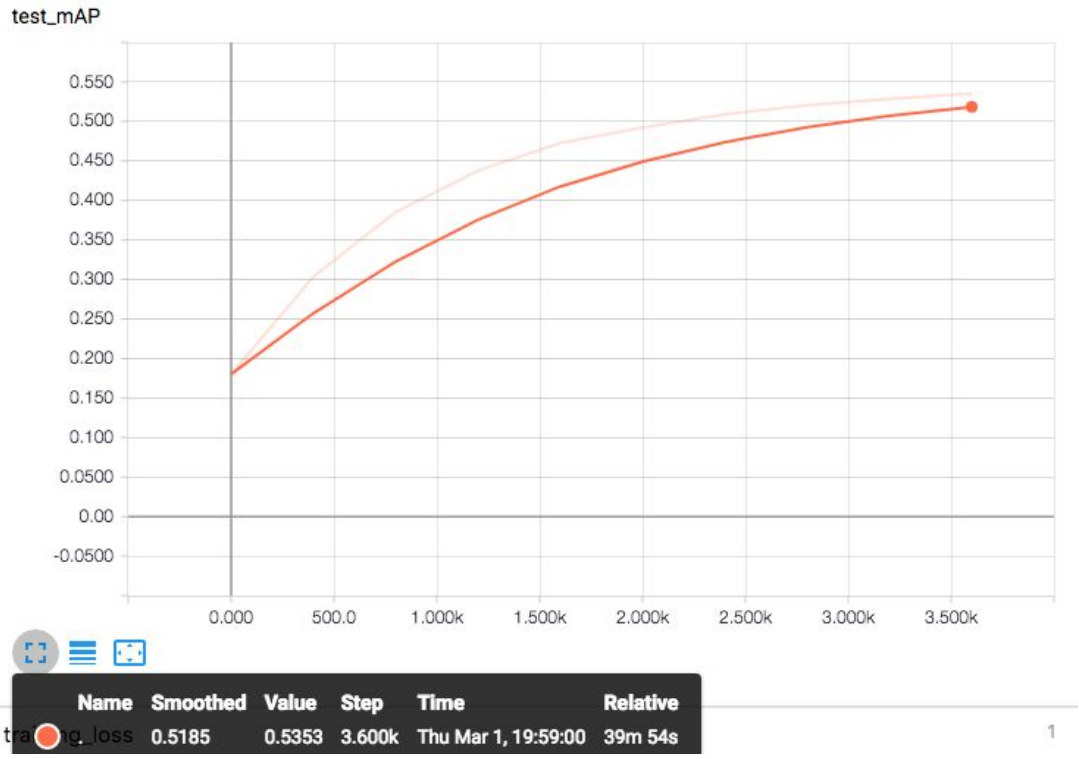
Final results for all classes:

```
INFO:tensorflow:Restoring parameters from ./tmp/vgg_model_scratch/model.ckpt-10400
Random AP: 0.0730556269628 mAP
GT AP: 1.0 mAP
Obtained 0.26245261537 mAP
per class:
aeroplane: 0.471582910336
bicycle: 0.118198403272
bird: 0.155353438978
boat: 0.259542243127
bottle: 0.128724894515
bus: 0.127580574914
car: 0.503521500351
cat: 0.249309898595
chair: 0.27290825276
cow: 0.151517572139
diningtable: 0.240832575523
dog: 0.231673580469
horse: 0.508845114497
motorbike: 0.250465864647
person: 0.68767489944
pottedplant: 0.1057704836
sheep: 0.184589659258
sofa: 0.184120680368
train: 0.304786973533
tvmonitor: 0.112052787075
```

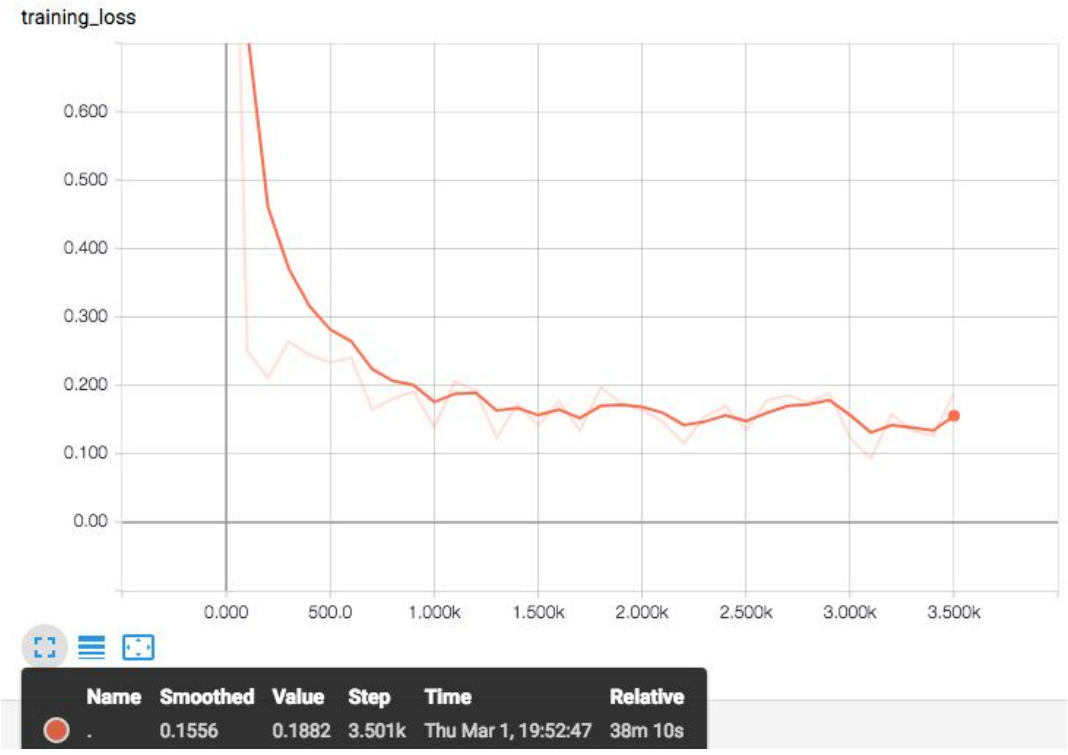
TASK4: Finetuning from ImageNet

(1) If initialize fc8 using zeros





Test mAP: **0.5353**

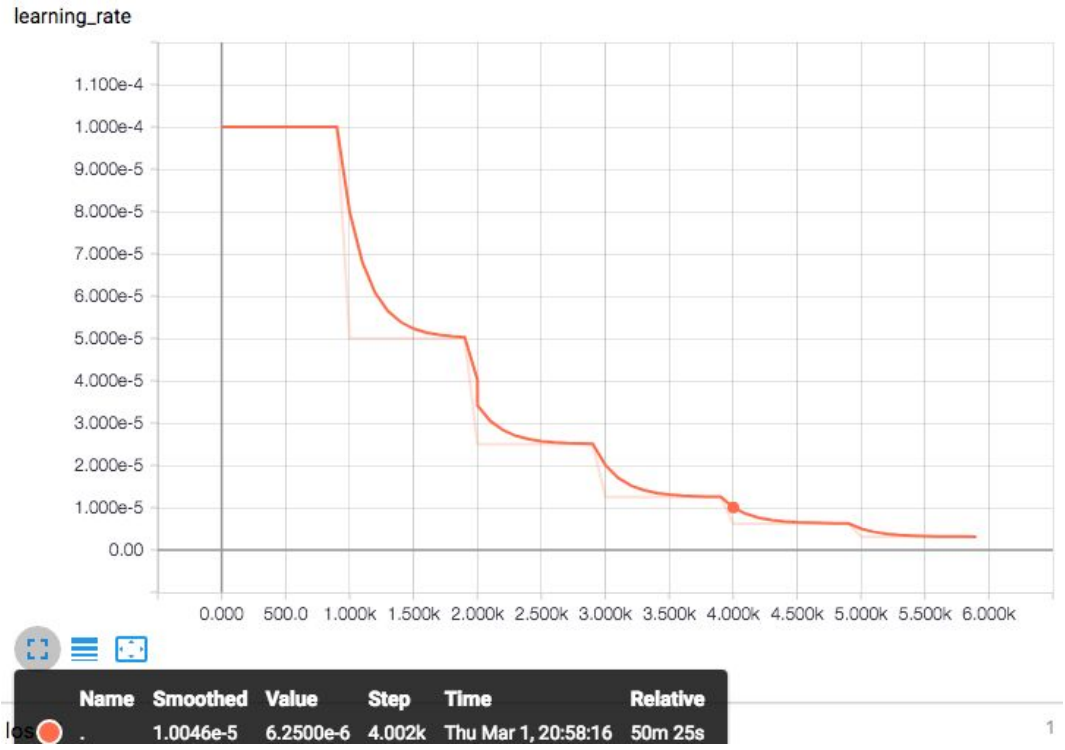


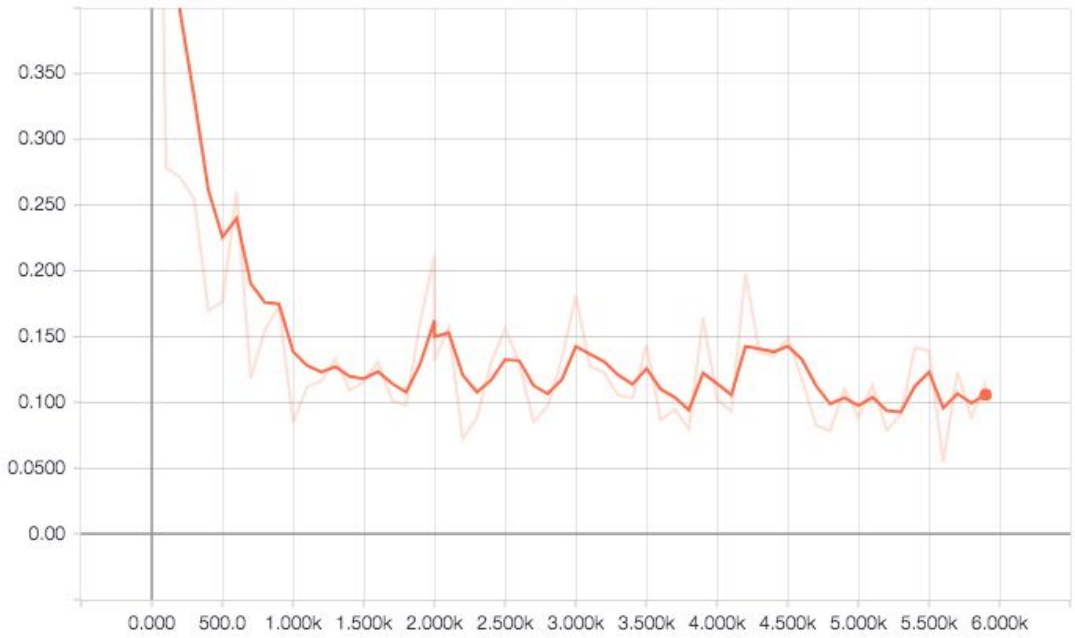
Train loss: **0.1882**

Final results for all classes:

```
INFO:tensorflow:Restoring parameters from ./tmp/vgg_model_scratch_finetune/model.ckpt-4000
Random AP: 0.0726345651206 mAP
GT AP: 1.0 mAP
Obtained 0.535306835557 mAP
per class:
aeroplane: 0.766121998483
bicycle: 0.547868192741
bird: 0.666774164297
boat: 0.437382216833
bottle: 0.193296576032
bus: 0.517461687135
car: 0.801330807193
cat: 0.613022913023
chair: 0.460073062025
cow: 0.311017041797
diningtable: 0.391776972524
dog: 0.573670714865
horse: 0.574351570085
motorbike: 0.554405360131
person: 0.912621118917
pottedplant: 0.279646202713
sheep: 0.547292401268
sofa: 0.298245241348
train: 0.718523861049
tvmonitor: 0.541254608691
```

(2) If Initialize fc8 layer using **gaussion**:






```
INFO:tensorflow:Restoring parameters from ./
Random AP: 0.0738082138784 mAP
GT AP: 1.0 mAP
Obtained 0.725994171054 mAP
per class:
aeroplane: 0.895064277365
bicycle: 0.812727749867
bird: 0.903309963583
boat: 0.856029621009
bottle: 0.339530325968
bus: 0.744409435519
car: 0.893394649743
cat: 0.856530908741
chair: 0.578549601755
cow: 0.492302294211
diningtable: 0.586374296466
dog: 0.838216807615
horse: 0.775817583656
motorbike: 0.755366937012
person: 0.948974045221
pottedplant: 0.474965430698
sheep: 0.677008321646
sofa: 0.575277130516
train: 0.889053404597
tvmonitor: 0.626980635884
accuracy: 0
Accuracy at iter 3600: 0.7259941710537019
```

Test mAP: **0.7260**

TASK5: Analysis

I include the code for visualization task in 02_AlexNet.py.

Visualize alexnet conv1 features

(1) At iter 601

(2) At iter 13301:

(3) At iter 22501:

(4) At iter 39901:

We could find that, as training process increases, the conv1 would learn more and more precise features of the image, such as color, edges etc.