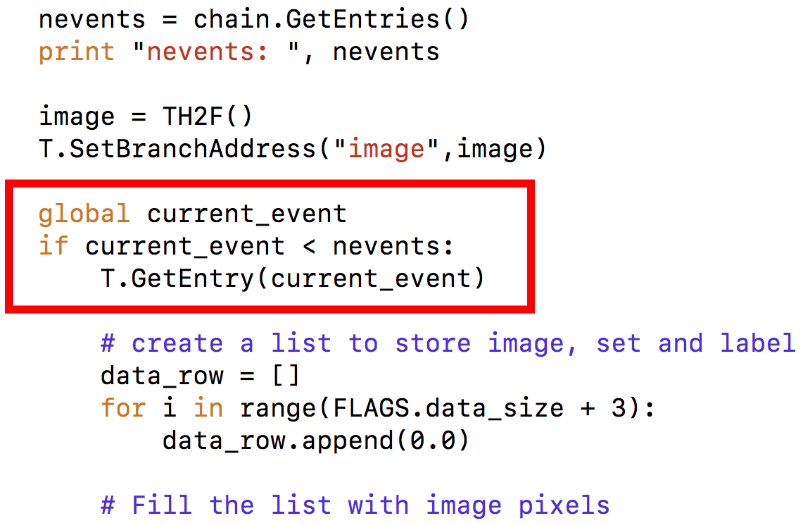
Progress Update

Enze Zhang

2017/09/07

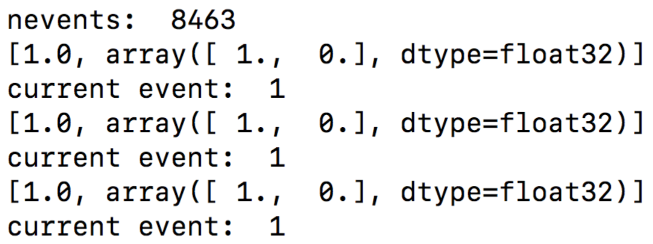
Difficulty in read_file function

- I use a global variable to go through events in the tree, and each time read in one event.



```
nevents = chain.GetEntries()
print "nevents: ", nevents

image = TH2F()
T.SetBranchAddress("image",image)

global current_event
if current_event < nevents:
    T.GetEntry(current_event)

    # create a list to store image, set and label
    data_row = []
    for i in range(FLAGS.data_size + 3):
        data_row.append(0.0)

    # Fill the list with image pixels
```

- But when I run a session multiple times, it will only call the function once and repeat the outputs.



```
nevents:  8463
[1.0, array([ 1.,  0.], dtype=float32)]
current event:  1
[1.0, array([ 1.,  0.], dtype=float32)]
current event:  1
[1.0, array([ 1.,  0.], dtype=float32)]
current event:  1
```

Difficulty in read_file function

- I did a few tests to clarify the problem:
  1) call the read_file function explicitly when run a session
  2) substitute the global variable with a queue object, which dequeues a different current_event number each time
  3) try the same thing with tf.TextLineReader
- Results:
  - 1) It worked, but I cannot do that in the whole program since the session only runs once for a batch, not a single event.
  - 2) It failed, because the output of a queue is a tensor, while GetEntry(i) only accepts an integer as input.
  - 3) The read function is still called only once, but the outputs are different.

Difficulty in read_file function

- So I wonder how tf.TextLineReader realize the function.

- tf.TextLineReader definition:
    - https://github.com/tensorflow/tensorflow/blob/r1.3/tensorflow/python/ops/io_ops.py
- source:
    - https://github.com/tensorflow/tensorflow/blob/r1.3/tensorflow/core/kernels/text_line_reader_op.cc

- But I don’ t really understand them.

An alternative method

- I wrote a small program *write.C* to write down all the events in a text file and shuffled them, and then read them in through tf.TextLineReader.

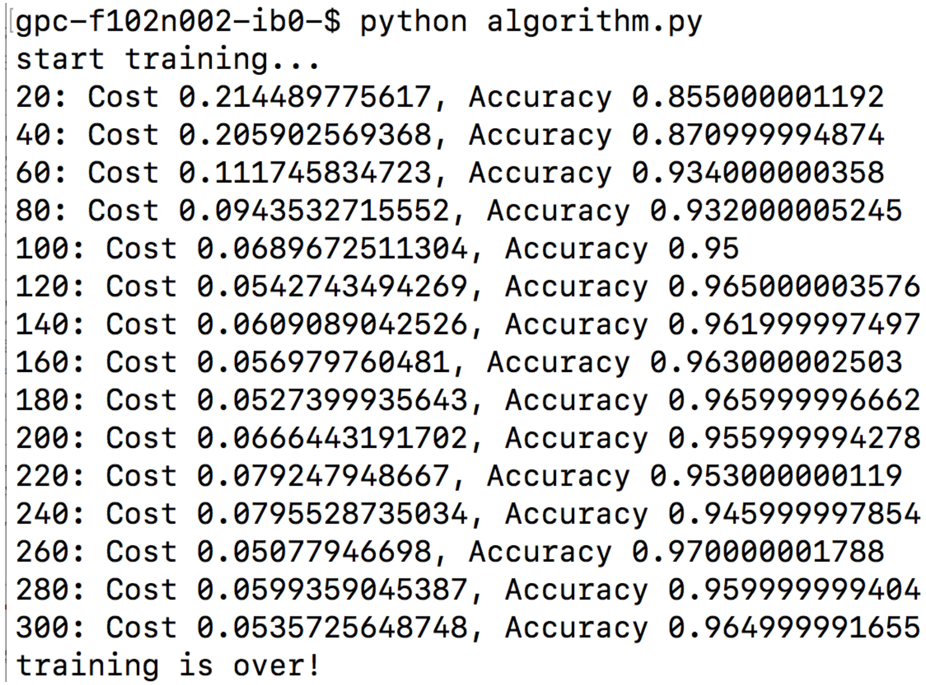
- Then I put the previous problem aside and continue to do training.

Training

- *algorithm.py* : a simplified version of Theo’ s algorithm, with no distinction between data sets.

- 300 batches, 50 events for each

- The accuracy is stable around 96%



```
gpc-f102n002-ib0-$ python algorithm.py
start training...
20: Cost 0.214489775617, Accuracy 0.855000001192
40: Cost 0.205902569368, Accuracy 0.870999994874
60: Cost 0.111745834723, Accuracy 0.934000000358
80: Cost 0.0943532715552, Accuracy 0.932000005245
100: Cost 0.0689672511304, Accuracy 0.95
120: Cost 0.0542743494269, Accuracy 0.965000003576
140: Cost 0.0609089042526, Accuracy 0.961999997497
160: Cost 0.056979760481, Accuracy 0.963000002503
180: Cost 0.0527399935643, Accuracy 0.965999996662
200: Cost 0.0666443191702, Accuracy 0.955999994278
220: Cost 0.079247948667, Accuracy 0.953000000119
240: Cost 0.0795528735034, Accuracy 0.945999997854
260: Cost 0.05077946698, Accuracy 0.970000001788
280: Cost 0.0599359045387, Accuracy 0.959999999404
300: Cost 0.0535725648748, Accuracy 0.964999991655
training is over!
```

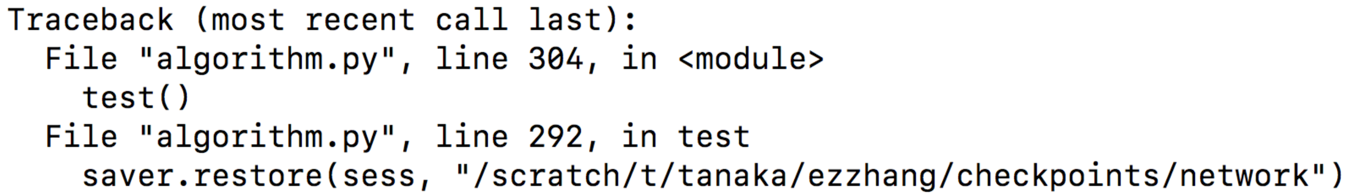
Test

- At first, I tried do a test right after training in the same session. However, it turned out they ran in parallel so that it didn’ t use the trained network for test.

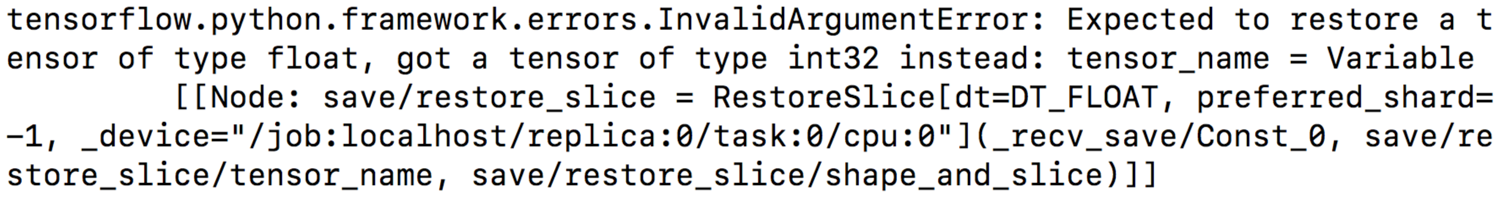
- So I do the test separately from training. In this case, I have to save the variables with tf.train.Saver after training, and restore them before testing.

Test

- But an error occurs when I restore the variables, and so far I have no idea what’s wrong.



```
Traceback (most recent call last):
  File "algorithm.py", line 304, in <module>
    test()
  File "algorithm.py", line 292, in test
    saver.restore(sess, "/scratch/t/tanaka/ezzhang/checkpoints/network")
```



```
tensorflow.python.framework.errors.InvalidArgumentError: Expected to restore a t
ensor of type float, got a tensor of type int32 instead: tensor_name = Variable
         [[Node: save/restore_slice = RestoreSlice[dt=DT_FLOAT, preferred_shard=
-1, _device="/job:localhost/replica:0/task:0/cpu:0"](_recv_save/Const_0, save/re
store_slice/tensor_name, save/restore_slice/shape_and_slice)]]
```

Next Step

- After the algorithm can run successfully, I will train the network separately for different data sets, and compare with Emily’ s results.