1.提供对densenet实现过程的描述： 对growth的理解 ， 对稠密链接的理解

想实现一个比较简单的densenet看看效果，就打算用3个block，分别是2,4,6层，growth为32

每个block都是稠密连接，

```
def block(net, layers, growth, scope='block'):
  for idx in range(layers):
    bottleneck = bn_act_conv_drp(net, 4 * growth, [1, 1],
                                 scope=scope + '_conv1x1' + str(idx))
    tmp = bn_act_conv_drp(bottleneck, growth, [3, 3],
                          scope=scope + '_conv3x3' + str(idx))
    net = tf.concat(axis=3, values=[net, tmp])
  return net
```

block之间用连接层transition连接，transition由1*1卷积，2*2平均池化组成

```
def transition(net, num_outputs, scope='transition'):
  net = bn_act_conv_drp(net, num_outputs, [1, 1], scope=scope + '_conv1x1')
  net = slim.avg_pool2d(net, [2, 2], stride=2, scope=scope + '_avgpool')
  return net
```

DenseNet核心思想在于建立了不同层之间的连接关系，充分利用了feature，进一步减轻了梯度消失问题，加深网络不是问题，

而且训练效果非常好。另外，利用bottleneck layer，Translation layer以及较小的growth rate使得网络变窄，参数减少，

有效抑制了过拟合，同时计算量也减少了。DenseNet优点很多，而且在和ResNet的对比中优势还是非常明显的。

2.开始训练模型https://www.tinymind.com/executions/4egw7rxo

在https://github.com/liqiang2018/quiz-w7-2-densenet 上完成densenet 网络后，通过tinymind开始训练

在 **载入点**输入 train_image_classifier.py

数据集 勾选 /data/ai100/quiz-w7/quiz_train_00000of00004.tfrecord

参数：

参数

iterations	learning_rate	batch_size	dropout	decay	output_dir
500	0.1	32	0.5	0.1	/output

dataset_name	dataset_dir	dataset_split_name	train_dir
quiz	/data/ai100/quiz-w7	train	/output/ckpt

model_name	optimizer	clone_on_cpu	eval_dir	max_num_batches
densenet	rmsprop	true	/output/eval	128

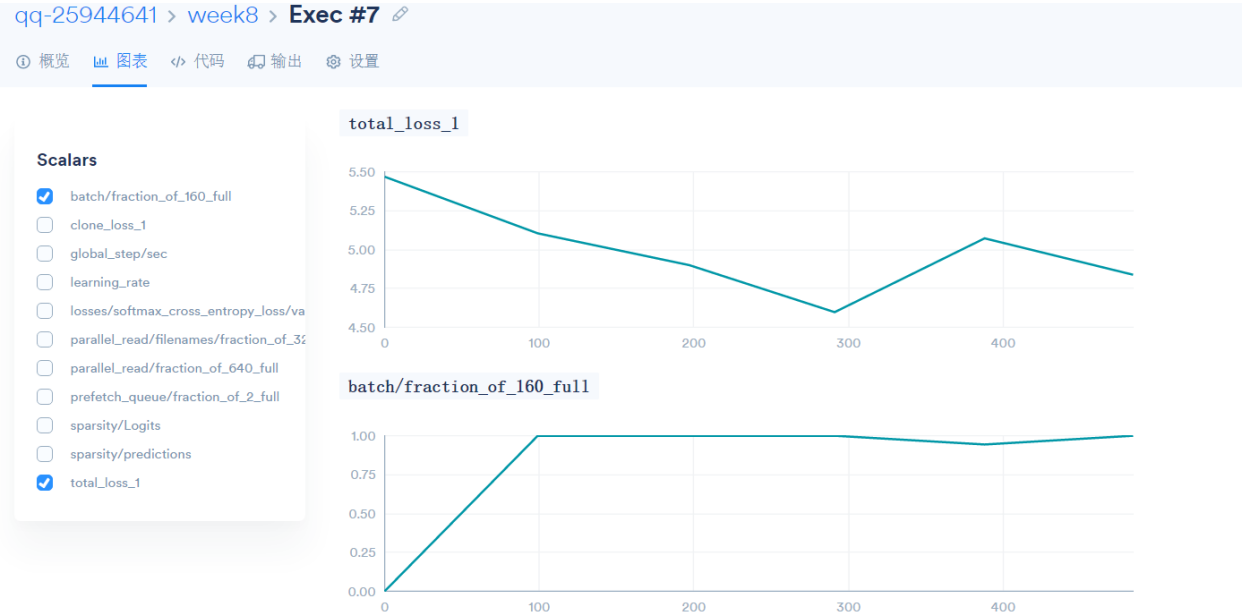
训练完成后

日志：

```
INFO:tensorflow:global step 580: loss = 4.9900 (7.706 sec/step)
INFO:tensorflow:Recording summary at step 580.
INFO:tensorflow:global step 590: loss = 4.8749 (6.228 sec/step)
INFO:tensorflow:global step 600: loss = 4.8882 (6.145 sec/step)
INFO:tensorflow:global step 610: loss = 4.7453 (6.182 sec/step)
INFO:tensorflow:global step 620: loss = 4.7620 (6.323 sec/step)
INFO:tensorflow:global step 630: loss = 5.0217 (6.303 sec/step)
INFO:tensorflow:global step 640: loss = 4.7714 (6.539 sec/step)
INFO:tensorflow:global step 650: loss = 4.9791 (6.840 sec/step)
INFO:tensorflow:global step 660: loss = 4.6820 (6.541 sec/step)
```

训练660次，提示没钱就终止运行了

图标信息



输出：

output / **ckpt**

文件夹中有 18 个文件	
checkpoint	343 B
events.out.tfevents.1527407837.95ca0e7d1975	8.46 MB
graph.pbtxt	3.55 MB
model.ckpt-196.data-00000-of-00001	8.69 MB
model.ckpt-196.index	11.23 KB
model.ckpt-196.meta	1.55 MB
model.ckpt-291.data-00000-of-00001	8.69 MB
model.ckpt-291.index	11.23 KB

3.利用训练好的模型来预测https://www.tinymind.com/executions/4ptbgv6j

在 **载入点**输入 eval_image_classifier.py来验证

虽然最后运行成功，并有正确率输出，但结果很让人失望

```
INFO:tensorflow:Evaluation [126/128]
INFO:tensorflow:Evaluation [127/128]
INFO:tensorflow:Evaluation [128/128]
2018-05-27 10:00:43.705997: I tensorflow/core/kernels/logging_ops.cc:79] eval/Accuracy[0.0124511719]
2018-05-27 10:00:43.705997: I tensorflow/core/kernels/logging_ops.cc:79] eval/Recall_5[0.0520019531]
INFO:tensorflow:Finished evaluation at 2018-05-27-10:00:43
```

但拥有的点数以用完，还是自己有充了5美元，才跑出的结果