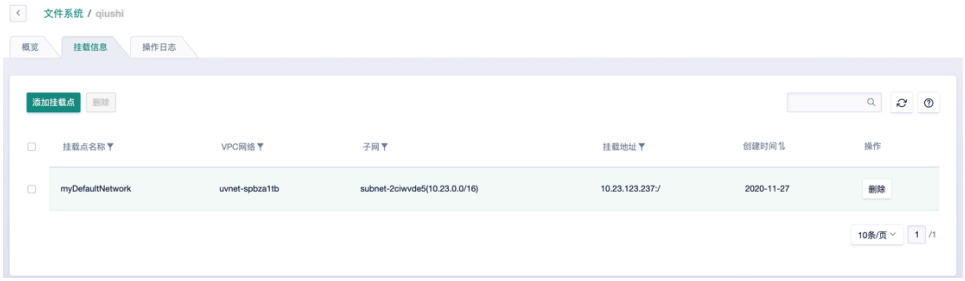
实验七报告

10185501402-孙秋实

作业 1：



作业 2：

```
[root@10-23-92-83 ~]# sudo mount -t nfs4 10.23.123.237:/ /mnt
[root@10-23-92-83 ~]# df -hT
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  399M     0  399M   0% /dev
tmpfs           tmpfs     412M     0  412M   0% /dev/shm
tmpfs           tmpfs     412M   11M  401M   3% /run
tmpfs           tmpfs     412M     0  412M   0% /sys/fs/cgroup
/dev/vda1       xfs        20G  1.8G   19G   9% /
/dev/vdb        xfs        20G   33M   20G   1% /data
tmpfs           tmpfs      83M     0   83M   0% /run/user/0
10.23.123.237:/ nfs4      100G     0  100G   0% /mnt
[root@10-23-92-83 ~]#
```

作业 3：

```
[root@10-23-92-83 mnist]# ls -la
total 25
drwxr-xr-x 6 root root 4096 Nov 27 08:36 .
drwxr-xr-x 3 root root 4096 Nov 27 08:34 ..
drwxr-xr-x 8 root root 4096 Nov 27 08:36 .git
-rw-r--r-- 1 root root   28 Nov 27 08:36 README.md
drwxr-xr-x 2 root root 4096 Nov 27 08:36 code
drwxr-xr-x 2 root root 4096 Nov 27 08:36 data
drwxr-xr-x 2 root root 4096 Nov 27 08:36 output
[root@10-23-92-83 mnist]#
```

作业 4：

作业 5：

（optimizer='adam'）

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),    #拉伸图像成一维向量
  tf.keras.layers.Dense(128, activation='relu'),    #第一层全连接+ReLU激活
  tf.keras.layers.Dropout(0.2),                     #dropout层
  tf.keras.layers.Dense(10, activation='softmax')   #第二层全连接+softmax激活，输出预测标签
])

#设置训练超参，优化器为sgd，损失函数为交叉熵，训练衡量指标为accuracy
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

In [36]:
```
#开始训练，训练15个epoch，一个epoch代表所有图像计算一遍。每一个epoch能观察到训练精度的提升
model.fit(x_train, y_train, epochs=15)
model.evaluate(x_test,  y_test)
```

```
Epoch 1/15
60000/60000 [==============================] - 4s 72us/sample - loss: 0.2979 - accuracy: 0.9150
Epoch 2/15
60000/60000 [==============================] - 4s 62us/sample - loss: 0.1468 - accuracy: 0.9565
Epoch 3/15
60000/60000 [==============================] - 4s 62us/sample - loss: 0.1078 - accuracy: 0.9686
Epoch 4/15
60000/60000 [==============================] - 4s 62us/sample - loss: 0.0886 - accuracy: 0.9723
Epoch 5/15
60000/60000 [==============================] - 4s 62us/sample - loss: 0.0787 - accuracy: 0.9754
Epoch 6/15
60000/60000 [==============================] - 4s 63us/sample - loss: 0.0674 - accuracy: 0.9783
Epoch 7/15
60000/60000 [==============================] - 4s 63us/sample - loss: 0.0590 - accuracy: 0.9807
Epoch 8/15
60000/60000 [==============================] - 4s 62us/sample - loss: 0.0531 - accuracy: 0.9827
Epoch 9/15
60000/60000 [==============================] - 4s 63us/sample - loss: 0.0480 - accuracy: 0.9845
Epoch 10/15
60000/60000 [==============================] - 4s 62us/sample - loss: 0.0442 - accuracy: 0.9852
Epoch 11/15
60000/60000 [==============================] - 4s 62us/sample - loss: 0.0411 - accuracy: 0.9863
Epoch 12/15
60000/60000 [==============================] - 4s 63us/sample - loss: 0.0393 - accuracy: 0.9869
Epoch 13/15
60000/60000 [==============================] - 4s 63us/sample - loss: 0.0353 - accuracy: 0.9877
Epoch 14/15
60000/60000 [==============================] - 4s 63us/sample - loss: 0.0357 - accuracy: 0.9878
Epoch 15/15
60000/60000 [==============================] - 4s 62us/sample - loss: 0.0336 - accuracy: 0.9885
10000/10000 [==============================] - 0s 36us/sample - loss: 0.0701 - accuracy: 0.9815
```

Out[36]:
```
[0.07013593632176526, 0.9815]
```