**计算机体系结构**

**计算机测评报告**

**姓 名： 吴尚锡**

**学 号： 16281115**

**自然班级： 1603**

**2019年 3 月 6 日**

**目录**

[**一、** **测评内容** 2](#_Toc2773990)

[**a)** **测评内容综述** 2](#_Toc2773991)

[**b)** **CPU介绍** 3](#_Toc2773992)

[**c)** **GPU介绍** 3](#_Toc2773993)

[**d)** **内存介绍** 3](#_Toc2773994)

[**e)** **外部储存介绍** 3](#_Toc2773995)

[**f)** **其他功能简述** 3](#_Toc2773996)

[**二、** **测评硬件简介** 4](#_Toc2773997)

[**b)** **测试案例2配置情况** 4](#_Toc2773998)

[**三、** **测评方案介绍** 5](#_Toc2773999)

[**a)** **测试计划一** 5](#_Toc2774000)

[**b)** **测试计划二** 5](#_Toc2774001)

[**四、** **测评结果展示** 5](#_Toc2774002)

[**a)** **案例1测评结果** 5](#_Toc2774003)

[**b)** **案例2测评结果** 9](#_Toc2774004)

[**五、** **总结** 9](#_Toc2774005)

1. **测评内容**
   1. **测评内容综述**

本文主要针对计算机系同学使用习惯和常用实验等需求对两台目标机器进行系统的全面的测试。以方便同学们了解什么样的计算机更适合购买，或更加适配于什么样的工作环境。本文主要从CPU、GPU、内存、外存和其他硬件功能进行测试，主要测试内容分为对日常程序即CPU运行压力的测试和对高性能计算即GPU的运行能力进行测试。

最终测试结果以本文后部分的报告形式给出。

* 1. **CPU介绍**

CPU包括运算逻辑部件、[寄存器](https://baike.baidu.com/item/%E5%AF%84%E5%AD%98%E5%99%A8)部件和控制部件等。CPU在电脑中主要负责处理指令、执行操作、控制时间、处理数据这几项功能。CPU的性能会直接的影响到用户的日常使用体验，所以在一台电脑中CPU的地位是相当高的。

* 1. **GPU介绍**

GPU为电脑的图像处理器，GPU能够进行超大规模的并行运算，在机器学习领域、图像处理领域和区块链领域中发挥着不可替代的重要作用。

* 1. **内存介绍**

内存是计算机中重要的部件之一，它是与CPU进行沟通的桥梁。计算机中所有程序的运行都是在内存中进行的，因此内存的性能对计算机的影响非常大。内存(Memory)也被称为[内存储器](https://baike.baidu.com/item/%E5%86%85%E5%AD%98%E5%82%A8%E5%99%A8/834392)，其作用是用于暂时存放CPU中的运算数据，以及与[硬盘](https://baike.baidu.com/item/%E7%A1%AC%E7%9B%98/159825)等[外部存储器](https://baike.baidu.com/item/%E5%A4%96%E9%83%A8%E5%AD%98%E5%82%A8%E5%99%A8/4843180)交换的数据。只要计算机在运行中，CPU就会把需要运算的数据调到内存中进行运算，当运算完成后CPU再将结果传送出来，内存的运行也决定了计算机的稳定运行。

* 1. **外部储存介绍**

外储存器是指除[计算机](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA)内存及[CPU缓存](https://baike.baidu.com/item/CPU%E7%BC%93%E5%AD%98)以外的储存器，此类储存器一般断电后仍然能保存[数据](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE/5947370)。装机中现在流行使用SSD和HDD作为外部储存器。

* 1. **其他功能简述**

电脑中还有很多模块也会对电脑性能造成影响，例如网卡、蓝牙等外设、主板和电源等。

1. **测评硬件简介**

测评主要在我的笔记本电脑和台式电脑间进行，案例一为我的笔记本电脑环境配置情况，案例2为我的台式电脑环境配置情况。

* 1. **测试案例1配置情况**

该案例为Dell的旗舰系列XPS15（2016年版本）的配置。

|  |  |
| --- | --- |
| 环境 | 配置 |
| CPU | Intel i7 6700HD |
| GPU | GTX 960M |
| 内存 | 8G 2400Hz |
| 外存 | 256G SSD + 1T HDD |
| 系统 | Windows10 |

* 1. **测试案例2配置情况**

该案例为组装机，用于架设深度学习服务器。

|  |  |
| --- | --- |
| 环境 | 配置 |
| CPU | Intel i7 7700K |
| GPU | GTX 1080ti  + P106 |
| 内存 | 32G 2400Hz |
| 外存 | 256G SSD + 1T HDD |
| 系统 | Arch Linux |

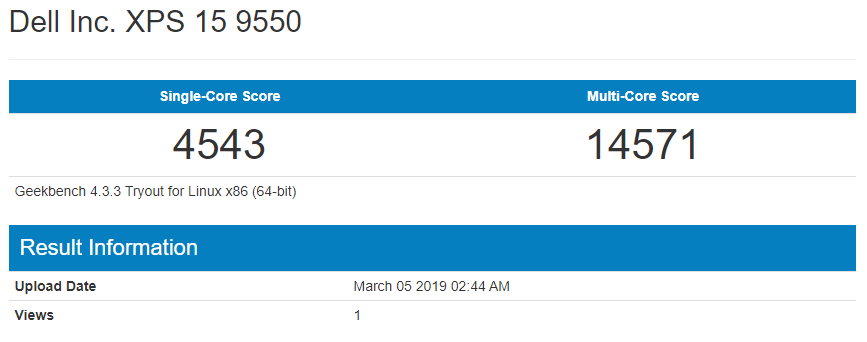
1. **测评方案介绍**
   1. **测试计划一**

使用开源软件Geekbench进行CPU性能测试，并将测试报告进行整理对比。分析出笔记本电脑和台式电脑的性能差距和适用范围。

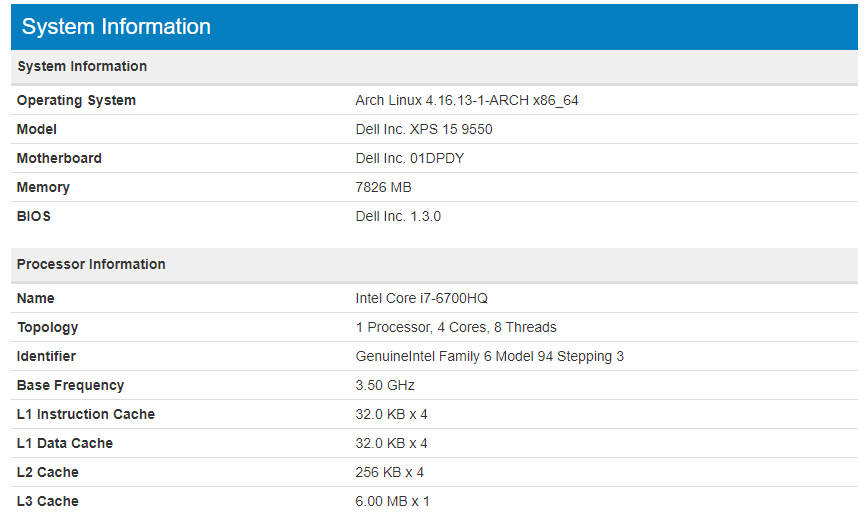
* 1. **测试计划二**

自行设计一段GPU训练代码和较长的C语言代码，测试机器GPU性能和CPU编译能力。综合考虑两台计算机对开发者的使用帮助。

1. **测评结果展示**
   1. **案例1测评结果**



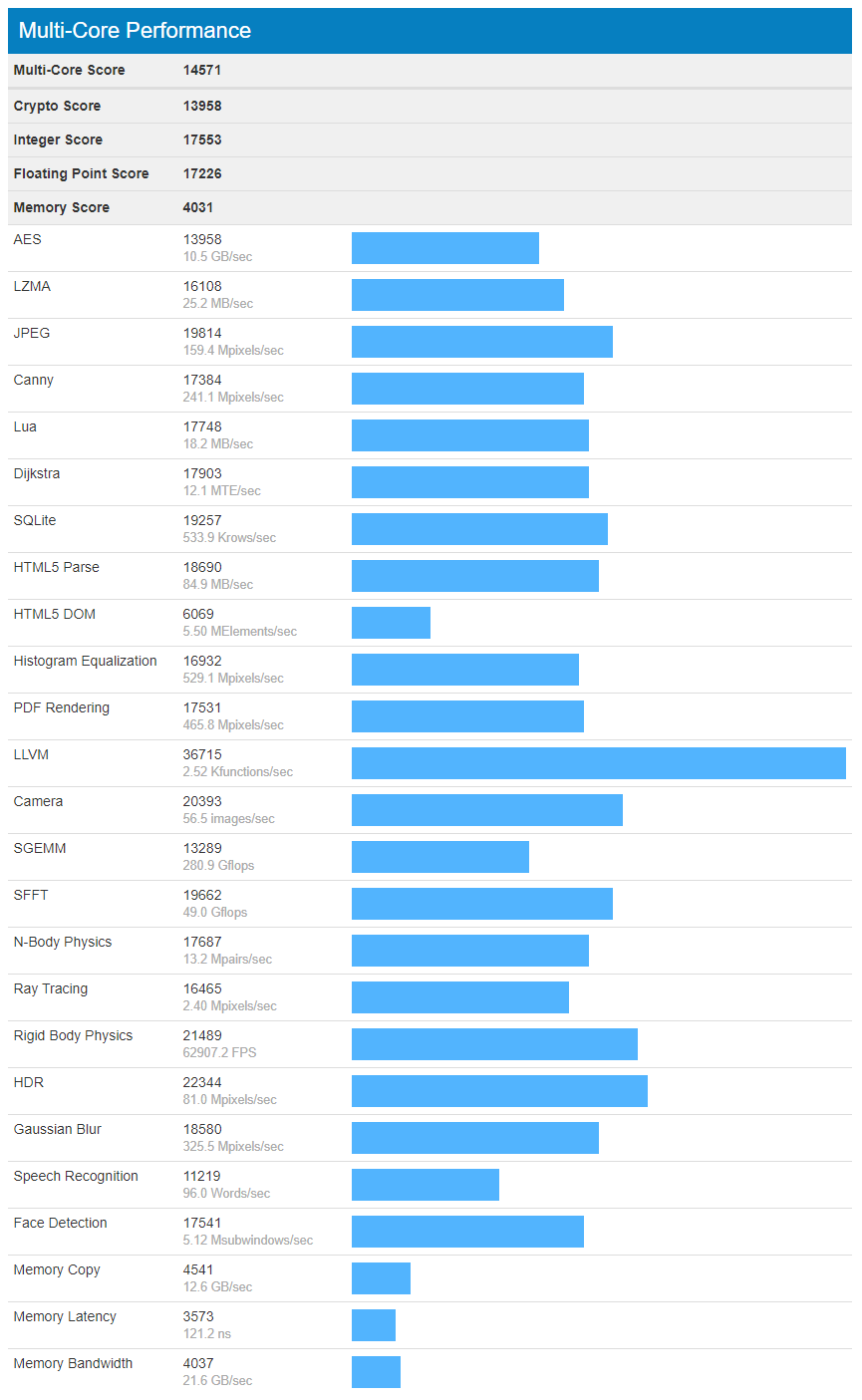
配置信息：



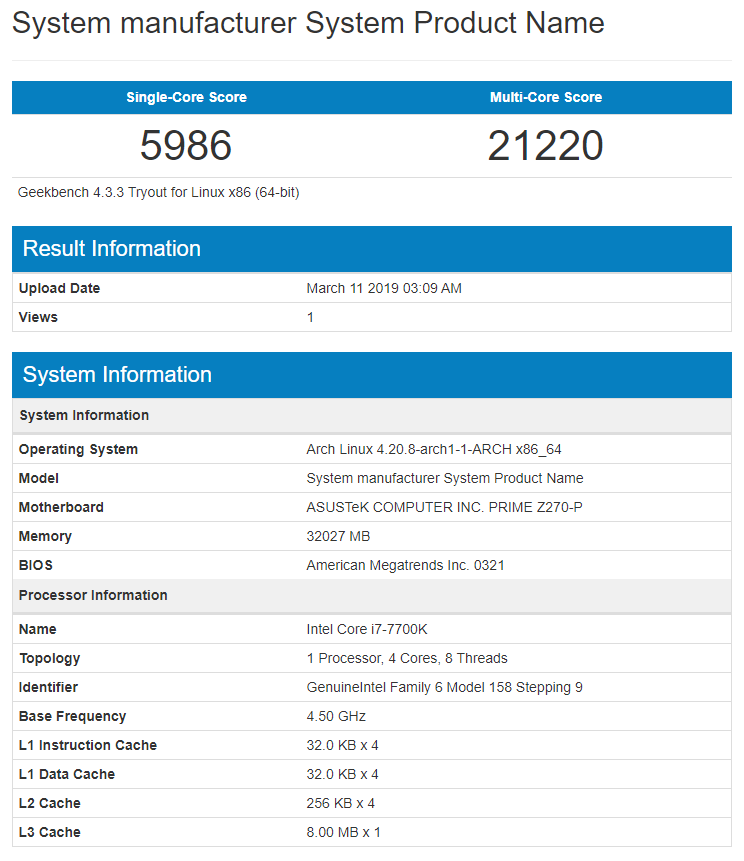
单核心处理能力：



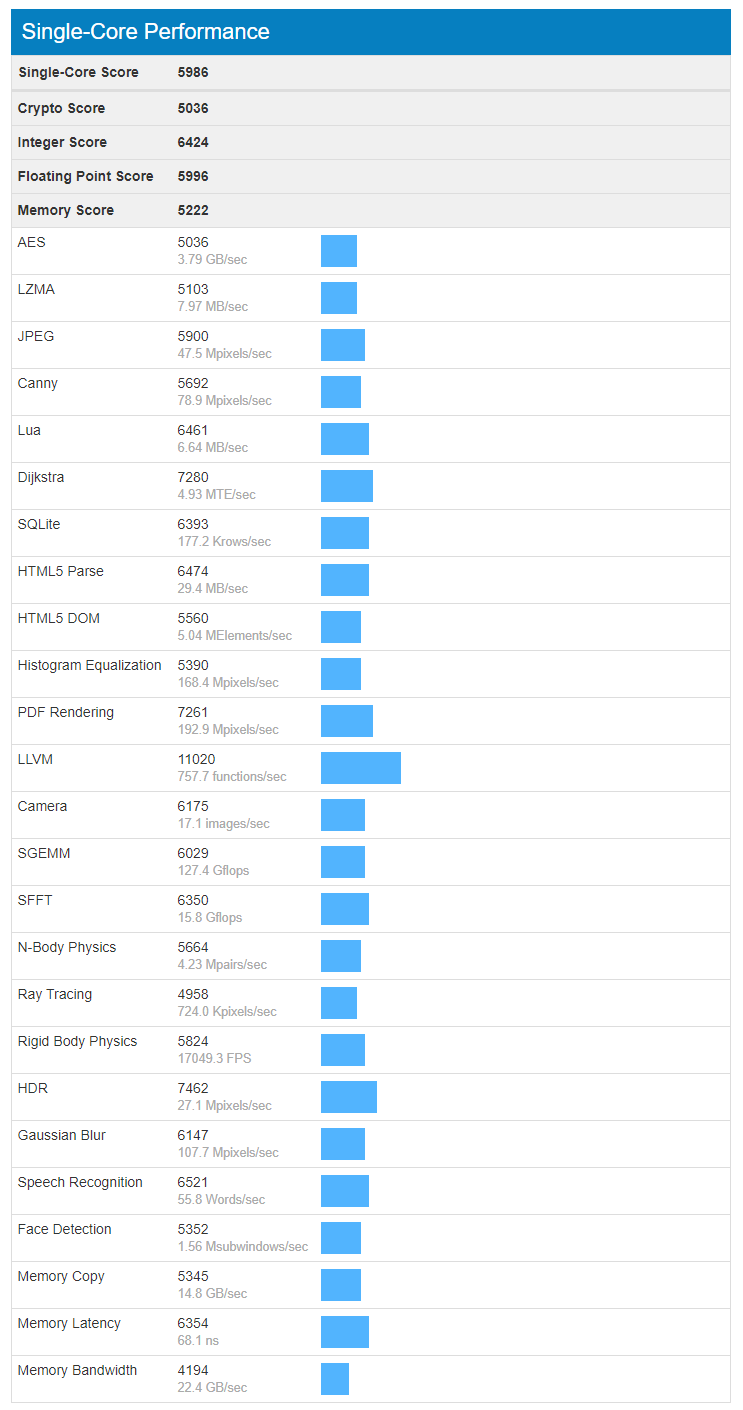
多核心处理能力：



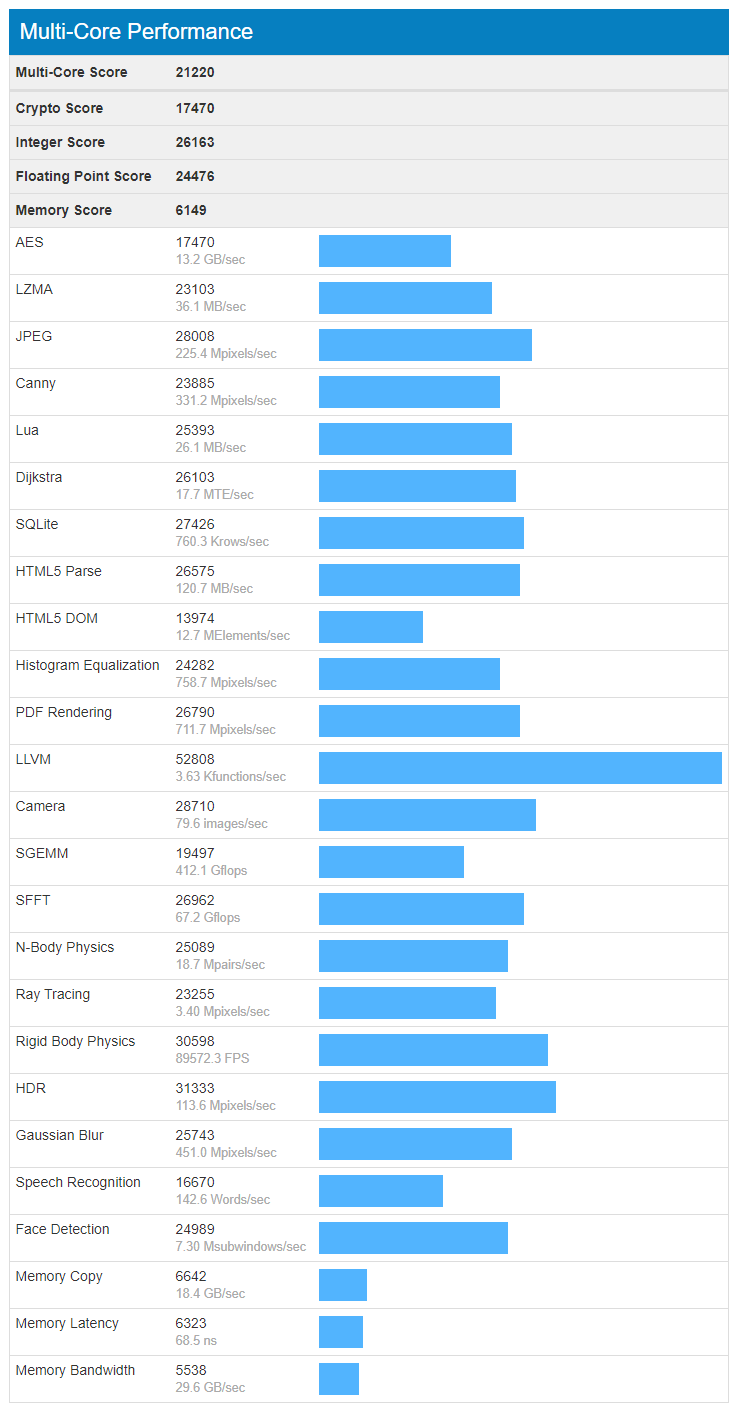
* 1. **案例2测评结果**



单核心处理能力



多核心处理能力



* 1. **测试二测评结果**

由于测试一主要测试CPU的运算能力，考虑到现在很多计算机系的同学会做机器学习和区块链的工作，我们需要对GPU做同样的测试。所以我设计了一个MNIST手写体训练的算法用于测试CPU能力。

该测试用例的程序代码如下：

import input\_data

import tensorflow as tf

import numpy as np

import datetime

starttime = datetime.datetime.now()

# 数据准备

mnist = input\_data.read\_data\_sets("MNIST\_data/", one\_hot=True)

X\_train = mnist.train.images

Y\_train = mnist.train.labels

xs = tf.placeholder(tf.float32, [None, 784])

ys = tf.placeholder(tf.float32, [None, 10])

# 搭建网络

# 第一卷积层

W\_conv1 = tf.Variable(tf.random\_normal([5, 5, 1, 32], stddev=0.1))

b\_conv1 = tf.Variable(tf.constant(0.1, shape=[32]))

conv\_1 = tf.nn.relu(tf.nn.conv2d(tf.reshape(

xs, [-1, 28, 28, 1]), W\_conv1, strides=[1, 1, 1, 1], padding='SAME')+b\_conv1)

pooling\_1 = tf.nn.max\_pool(conv\_1, ksize=[1, 2, 2, 1], strides=[

1, 2, 2, 1], padding='SAME') # 13\*13

# 第二卷积层

W\_conv2 = tf.Variable(tf.random\_normal([5, 5, 32, 64], stddev=0.1))

b\_conv2 = tf.Variable(tf.constant(0.1, shape=[64]))

conv\_2 = tf.nn.relu(tf.nn.conv2d(pooling\_1, W\_conv2, strides=[

1, 1, 1, 1], padding='SAME')+b\_conv2)

pooling\_2 = tf.nn.max\_pool(conv\_2, ksize=[1, 2, 2, 1], strides=[

1, 2, 2, 1], padding='SAME') # 7\*7

# 全连接层1

W\_3 = tf.Variable(tf.random\_normal([7\*7\*64, 1024], stddev=0.1))

b\_3 = tf.Variable(tf.constant(0.1, shape=[1024]))

l\_3\_0 = tf.reshape(pooling\_2, [-1, 7\*7\*64])

l\_3\_1 = tf.matmul(l\_3\_0, W\_3)+b\_3

L\_3 = tf.nn.relu(l\_3\_1)

# 全连接层2

W\_3 = tf.Variable(tf.random\_normal([1024, 1024], stddev=0.1))

b\_3 = tf.Variable(tf.constant(0.1, shape=[1024]))

L\_4 = tf.nn.sigmoid(tf.matmul(L\_3, W\_3)+b\_3)

# dropout

keep\_prob = tf.placeholder("float")

h\_fc1\_drop = tf.nn.dropout(L\_4, keep\_prob)

# output

W\_out = tf.Variable(tf.random\_normal([1024, 10], stddev=0.1))

b\_out = tf.Variable(tf.constant(0.1, shape=[10]))

y\_out = tf.nn.softmax(tf.matmul(h\_fc1\_drop, W\_out)+b\_out)

# 训练过程

loss = -tf.reduce\_sum(ys\*tf.log(y\_out))

train\_step = tf.train.GradientDescentOptimizer(0.00005).minimize(loss)

correct\_p = tf.equal(tf.argmax(y\_out, 1), (tf.argmax(ys, 1)))

accuracy = tf.reduce\_mean(tf.cast(correct\_p, "float"))

sess = tf.Session()

sess.run(tf.global\_variables\_initializer())

print("开始训练:")

for i in range(20000):

batch = mnist.train.next\_batch(50)

sess.run(train\_step, feed\_dict={

xs: batch[0], ys: batch[1], keep\_prob: 0.5})

if i % 2000 == 0:

print(i/2000)

print(sess.run(accuracy, feed\_dict={

xs: batch[0], ys: batch[1], keep\_prob: 1.0}))

print("进行测试集测试:")

testbatch = mnist.test.next\_batch(1000)

print(sess.run(accuracy, feed\_dict={

xs: testbatch[0], ys: testbatch[1], keep\_prob: 1.0}))

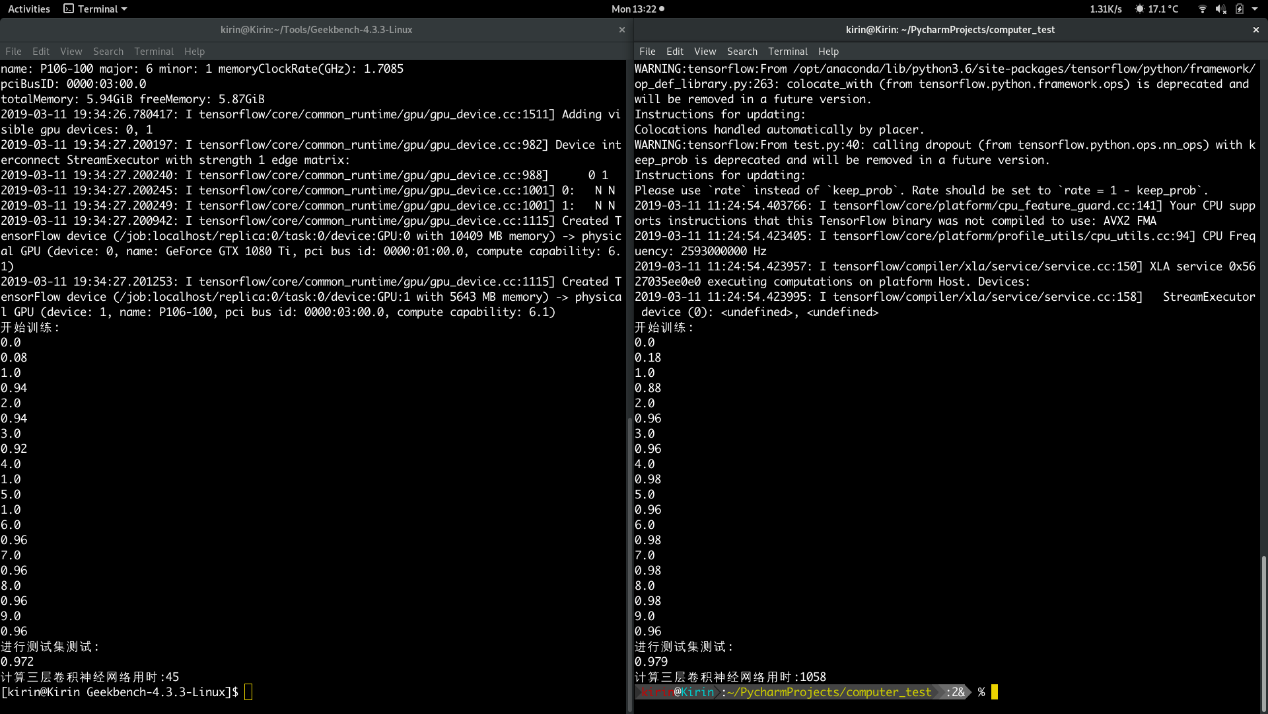
endtime = datetime.datetime.now()

print("计算三层卷积神经网络用时:{}".format((endtime - starttime).seconds))

测试结果如下：

右边为笔记本电脑用时，为1058s

左边为台式电脑用时，为45s



1. **总结**

通过系列测试，我给打算进军机器学习等领域的同学提出以下建议，建议使用轻便的笔记本电脑，并且购买或者租赁一个GPU服务器。笔记本电脑用于随身出行，而后台服务器用于计算服务。

因为GPU的运算能力远远强于CPU的计算能力，好的GPU会给模型调优带来极大的便利。