**2020年安徽省大数据与人工智能应用初赛**

**学校名称： 队伍名称： 队伍编号：**

# 第五部分 人工智能（20分）

作为新一轮产业变革的核心驱动力和引领未来发展的战略技术，国家高度重视人工智能产业的发展。2017年国务院发布《新一代人工智能发展规划》，对人工智能产业进行战略部署;在2018年3月和2019年3月的政府工作报告中，均强调指出要加快新兴产业发展，推动人工智能等研发应用，培育新一代信息技术等新兴产业集群壮大数字经济。下面请你完成人工智能技术框架TensorFLow 的安装及算法实现。

1. TensorFlow安装（6分）

1 请在linux环境下安装Python环境（3分）

导入安装包



解压文件夹



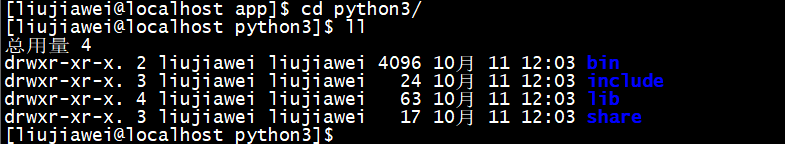
配置安装位置



编译并安装



编译结果



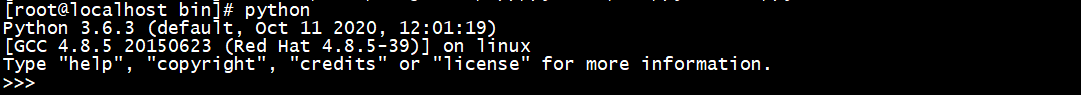
将原始python保存备份



建立python软连接



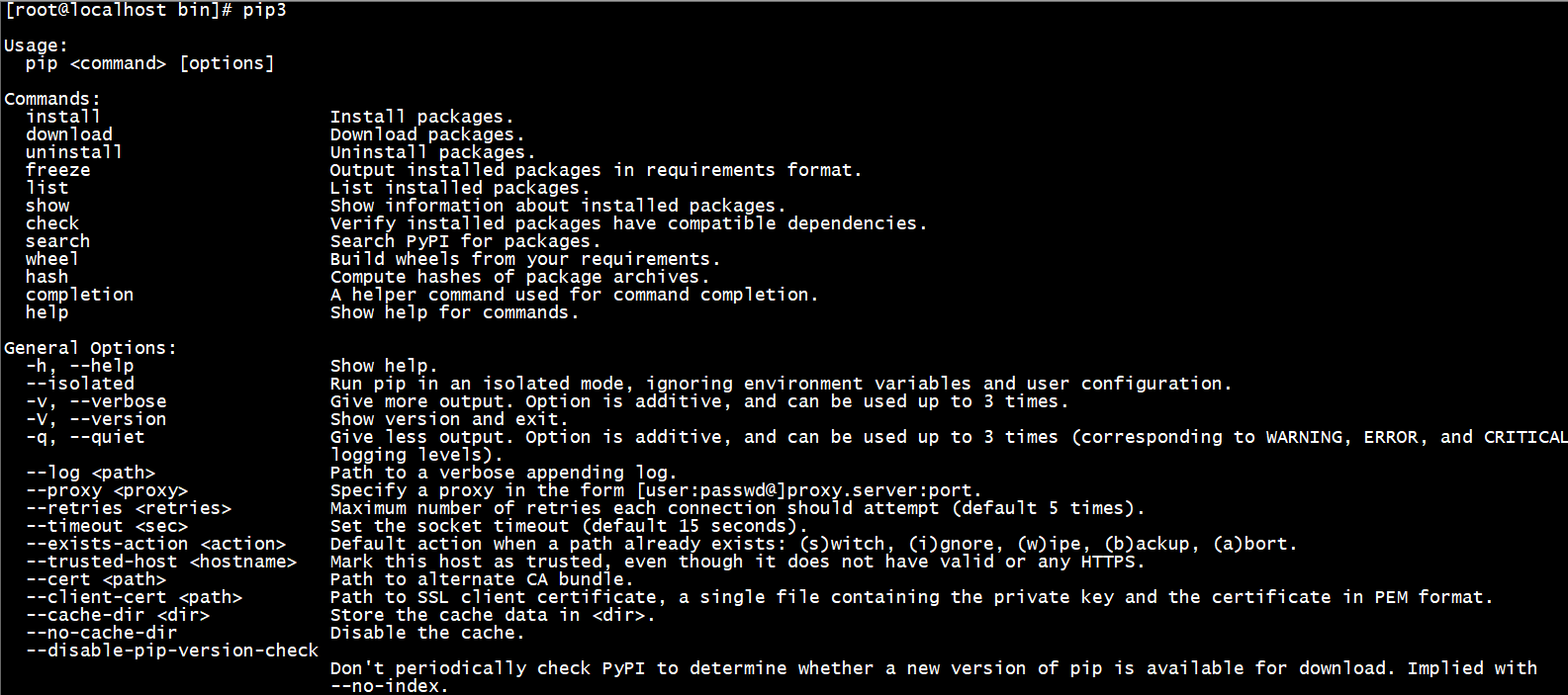
运行python



建立pip软连接

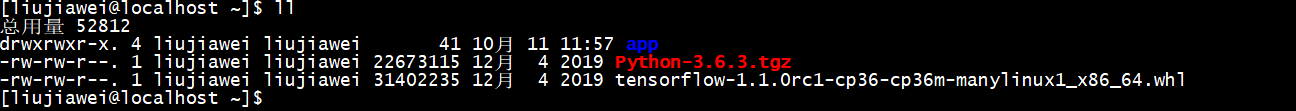


Pip结果：



2 请在linux下安装tensorflow（3分）

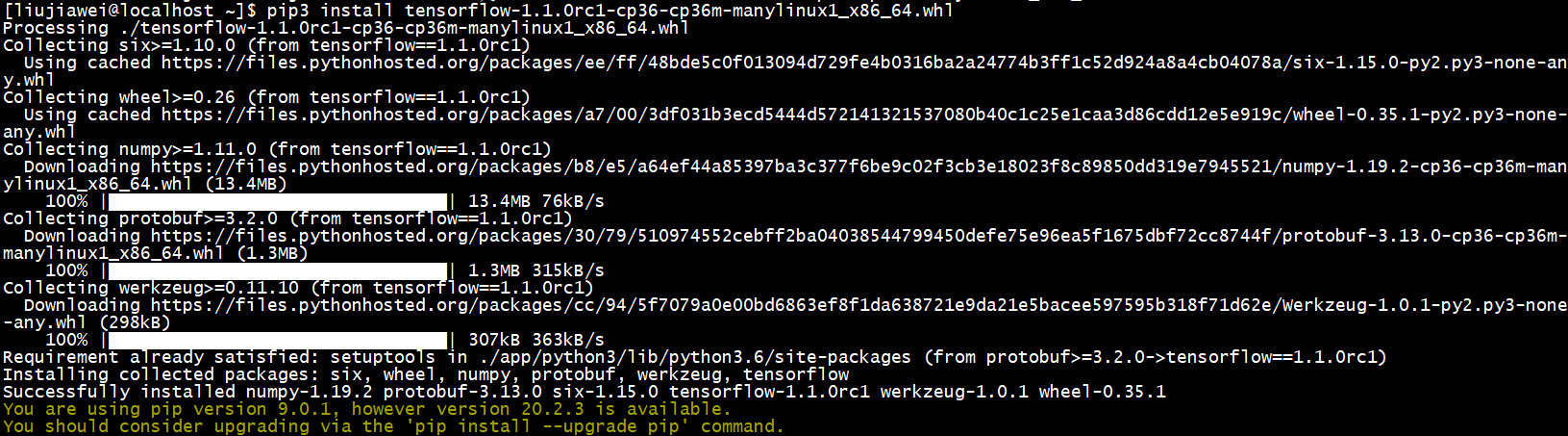
上传tensorflow的whl包



Pip安装



安装结果



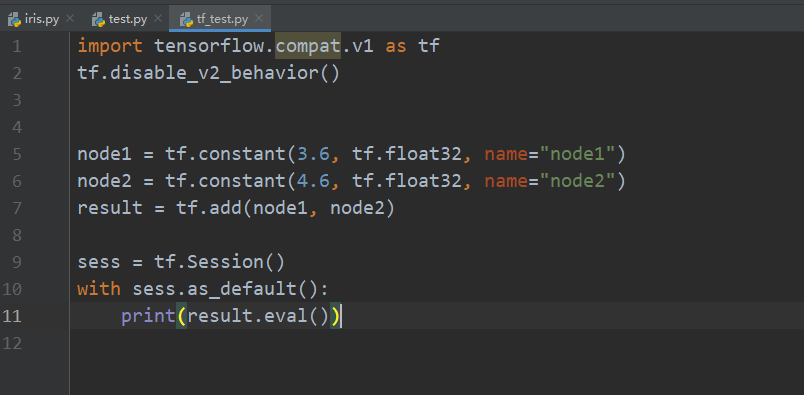
1. TensorFlow编程实现（14分）

1 请你构建一个简单的图计算（4分）

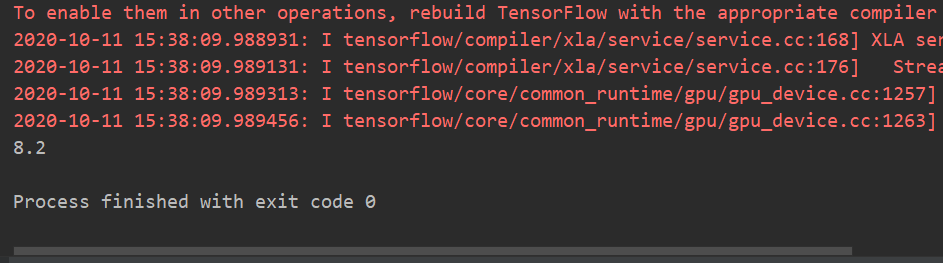
两个常量点（3.6）、（4.6），请你创建对两个 Tensor 执行 + 操作

请提供代码及截图（）

**代码：**



**结果：**

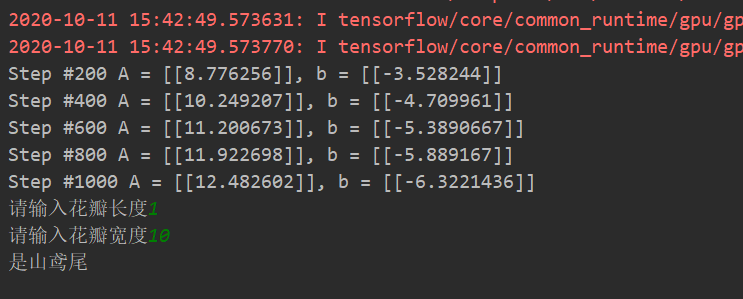


## 2 Iris数据集是常用的分类实验数据集，也称鸢尾花卉数据集，是一类多重变量分析的数据集。请你从python相关库中导入获取并使用tensorflow实现一个简单的二值分类器来预测一朵花是否为山鸢尾（提供相关代码和输出结果）（5分）

**代码：**

|  |
| --- |
| import matplotlib.pyplot as plt  import numpy as np  from sklearn import datasets  import tensorflow.compat.v1 as tf  from tensorflow.python.framework import ops  tf.disable\_v2\_behavior()  ops.reset\_default\_graph()  iris = datasets.load\_iris()  binary\_target = np.array([1. if x == 0 else 0. for x in iris.target])  iris\_2d = np.array([[x[2], x[3]] for x in iris.data])  # 声明批量训练大小  batch\_size = 20  # 初始化计算图  sess = tf.Session()  # 声明数据占位符  x1\_data = tf.placeholder(shape=[None, 1], dtype=tf.float32)  x2\_data = tf.placeholder(shape=[None, 1], dtype=tf.float32)  y\_target = tf.placeholder(shape=[None, 1], dtype=tf.float32)  # 声明模型变量  # Create variables A and b (0 = x1 - A\*x2 + b)  A = tf.Variable(tf.random\_normal(shape=[1, 1]))  b = tf.Variable(tf.random\_normal(shape=[1, 1]))  # 定义线性模型：  # 如果找到的数据点在直线以上，则将数据点代入x2-x1\*A-b计算出的结果大于0；  # 同理找到的数据点在直线以下，则将数据点代入x2-x1\*A-b计算出的结果小于0。  # x1 - A\*x2 + b  my\_mult = tf.matmul(x2\_data, A)  my\_add = tf.add(my\_mult, b)  my\_output = tf.subtract(x1\_data, my\_add)  # 增加TensorFlow的sigmoid交叉熵损失函数(cross entropy)  xentropy = tf.nn.sigmoid\_cross\_entropy\_with\_logits(logits=my\_output, labels=y\_target)  # 声明优化器方法  my\_opt = tf.train.GradientDescentOptimizer(0.05)  train\_step = my\_opt.minimize(xentropy)  # 创建一个变量初始化操作  init = tf.global\_variables\_initializer()  sess.run(init)  # 运行迭代1000次  for i in range(1000):  rand\_index = np.random.choice(len(iris\_2d), size=batch\_size)  # rand\_x = np.transpose([iris\_2d[rand\_index]])  # 传入三种数据：花瓣长度、花瓣宽度和目标变量  rand\_x = iris\_2d[rand\_index]  rand\_x1 = np.array([[x[0]] for x in rand\_x])  rand\_x2 = np.array([[x[1]] for x in rand\_x])  # rand\_y = np.transpose([binary\_target[rand\_index]])  rand\_y = np.array([[y] for y in binary\_target[rand\_index]])  sess.run(train\_step, feed\_dict={x1\_data: rand\_x1, x2\_data: rand\_x2, y\_target: rand\_y})  if (i + 1) % 200 == 0:  print('Step #' + str(i + 1) + ' A = ' + str(sess.run(A)) + ', b = ' + str(sess.run(b)))  # 绘图  # 获取斜率/截距  # Pull out slope/intercept  [[slope]] = sess.run(A)  [[intercept]] = sess.run(b)  # 创建拟合线  x = np.linspace(0, 3, num=50)  ablineValues = []  for i in x:  ablineValues.append(slope \* i + intercept)  # 对输入数据进行预测  input\_x1 = float(input('请输入花瓣长度'))  input\_x2 = float(input('请输入花瓣宽度'))  out\_y = input\_x2 - (slope \* input\_x1 + intercept)  if out\_y >= 0:  print("是山鸢尾")  else:  print('不是山鸢尾') |

**结果：**



3 请绘制分类器拟合曲线图（代码和结果图）（5分）

**代码（接上题）：**

# 绘图  
# 获取斜率/截距  
# Pull out slope/intercept  
[[slope]] = sess.run(A)  
[[intercept]] = sess.run(b)  
# 创建拟合线  
x = np.linspace(0, 3, num=50)  
ablineValues = []  
for i in x:  
 ablineValues.append(slope \* i + intercept)  
# 绘制拟合曲线  
plt.rcParams['font.sans-serif'] = ['SimHei'] # 用来正常显示中文标签  
setosa\_x = [a[1] for i, a in enumerate(iris\_2d) if binary\_target[i] == 1]  
setosa\_y = [a[0] for i, a in enumerate(iris\_2d) if binary\_target[i] == 1]  
non\_setosa\_x = [a[1] for i, a in enumerate(iris\_2d) if binary\_target[i] == 0]  
non\_setosa\_y = [a[0] for i, a in enumerate(iris\_2d) if binary\_target[i] == 0]  
plt.plot(setosa\_x, setosa\_y, 'rx', ms=10, mew=2, label='山鸢尾')  
plt.plot(non\_setosa\_x, non\_setosa\_y, 'ro', label='不是山鸢尾')  
plt.plot(x, ablineValues, 'b-')  
plt.xlim([0.0, 2.7])  
plt.ylim([0.0, 7.1])  
plt.suptitle('山鸢尾线性分类器', fontsize=20)  
plt.xlabel('花瓣长度')  
plt.ylabel('花瓣宽度')  
plt.legend(loc='lower right')  
plt.show()

**分类器拟合曲线图：**

