**人工智能模拟竞赛（一）**

# 项目背景

自深度学习的问世，它就在图像识别、目标检测、自然语言处理等领域取得巨大的成就。与此同时，TensorFlow、PyTorch等深度学习框架应运而生，以便更简单的将深度学习知识应用于实践中。因此，本项目依托TensorFlow深度学习框架，使用深度学习方法，通过使用深度卷积神经网络模型，实现菜品种类识别。

# 竞赛任务

本项目任务分为平台搭建、数据集加载及预处理、模型构建及训练、模型加载及预测、综合分析报告编写共计五个竞赛任务，请你们团队科学计划、合理分配，最终完成各阶段任务。

## 任务一 平台搭建

本项目中基于TensorFlow的深度学习训练环境需要GPU设备提供训练加速支持，因此，该竞赛任务需要安装NVIDIA显卡驱动、安装NVIDIA CUDA计算平台并验证、安装NVIDIA cuDNN深度学习加速平台并验证、编写TensorFlow GPU版的验证程序并进行验证。完成本任务需要的全部资源列表如下：

### 安装NVIDIA显卡驱动

完成显卡驱动(NVIDIA-Linux-x86\_64-418.43.run)的安装及验证工作，使用相关命令及操作完成NVIDIA显卡驱动的安装，安装结束后使用NVIDIA GPU状态监测命令nvidia-smi验证安装结果，在命令行终端输出GPU监测信息，并截图（shift+Print Screen）保存为“gpu安装验证结果.png”文件。

将验证结果截图文件保存在任务文件夹下。

### 安装NVIDIA CUDA计算平台并验证

完成NVIDIA CUDA计算平台(cuda\_9.0.176\_384.81.run)的安装及验证工作，使用相关命令及操作完成NVIDIA CUDA计算平台的安装，安装结束后使用相关命令验证安装结果，在命令行终端输出CUDA版本号，并截图（shift+Print Screen）保存为“cuda安装验证结果.png”文件。

将验证结果截图文件保存在桌面任务文件夹下。

### 安装NVIDIA cuDNN深度学习加速平台并验证

完成NVIDIA cuDNN加速平台的安装及验证工作，使用相关命令及操作完成NVIDIA cuDNN加速平台的安装，安装结束后使用相关命令验证cuDNN安装结果，在命令行终端输出cuDNN的测试结果，并截图（shift+Print Screen）保存为“cuDNN安装验证结果.png”文件。

将验证结果截图文件保存在任务文件夹下。

### 编写TensorFlow GPU版的验证程序并进行验证

编写TensorFlow GPU版可运行性验证代码保存为“TensorFlow可运行验证结果.png”文件。

将验证结果截图文件保存在桌面任务文件夹下。

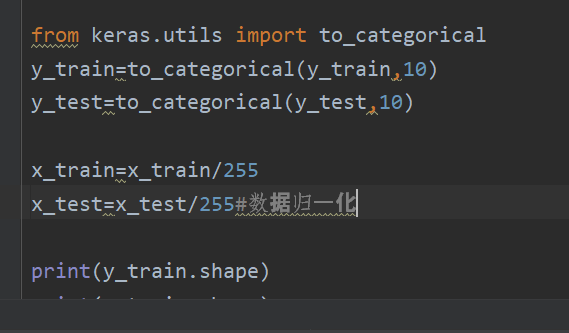
## 任务二 数据集加载及预处理

本次任务所用到的数据集为mnist手写数字数据集（此数据集keras自带，导入方法为from keras.datasets import mnist），为了图像数据集能够输入模型进行训练，项目组需要进行数据集加载及预处理工作。按照以下步骤完成数据集加载及预处理的相关代码。完成本任务需要的全部资源列表如下：

### 1.完成数据集预处理

#### 1.1图像数据归一化处理的代码实现

完成图像数据归一化处理并插入imgs”，并截图（shift+Print Screen）保存为“图像数据归一化.png”文件。

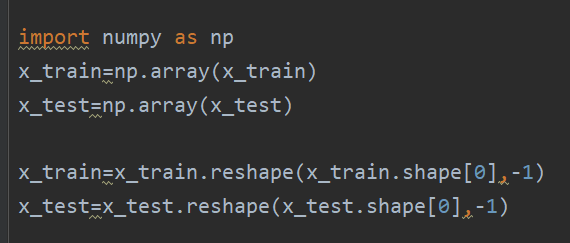


将代码截图文件保存在桌面文件夹下。

#### 1.2图像的数组转换处理的代码实现

完成图像的数组转换”并截图（shift+Print Screen）保存为“图像的数组转换处理.png”文件。

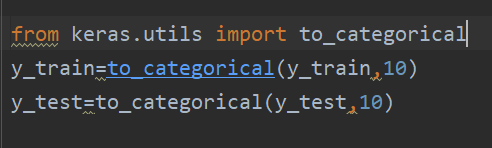
将代码截图文件保存在桌面任务文件夹下。



#### 1.3标签独热编码处理的代码实现

完成标签的独热编码处理”，并截图（shift+Print Screen）保存为“标签的独热编码处理.png”文件。

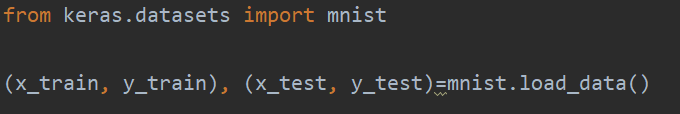
将代码截图文件保存在桌面任务文件夹下。



### 2.加载数据集

完成训练集、验证集、测试集的变量赋值”，并截图（shift+Print Screen）保存为“数据集载入.png”文件。

将代码截图文件保存在桌面任务文件夹下。



## 任务三 模型构建及训练

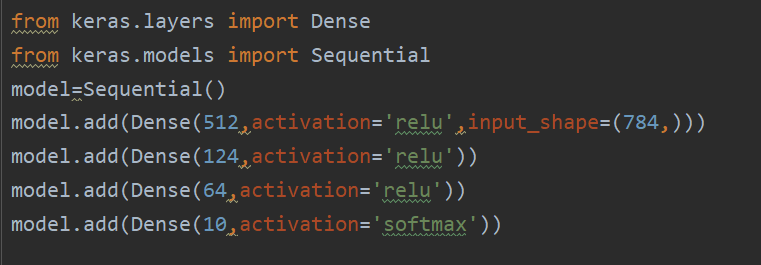
经过上一个任务的数据集预处理工作，下一步就可编写神经网络模型进行训练了，团队的研发经理已完成了主体的代码模板，按照以下步骤完成模型构建及训练的相关代码。完成本任务需要的全部资源列表如下：

### 完成模型构建

自行完成

并截图（shift+Print Screen）保存为“模型构建.png”文件。

将代码截图文件保存在桌面任务文件夹下。

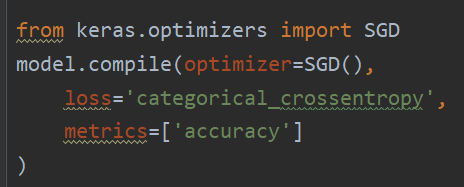


### 完成模型编译方法

自行完成

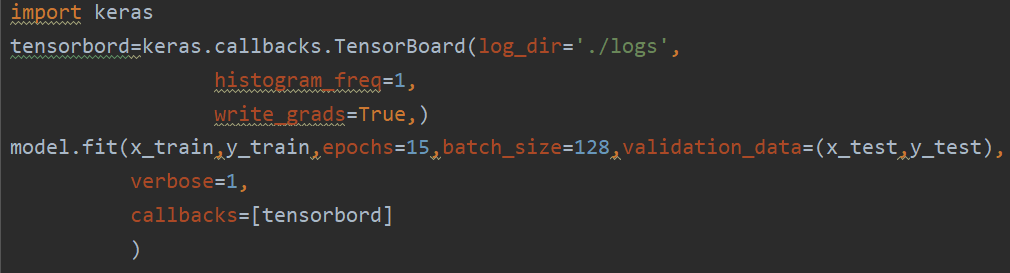
模型编译方法”并截图（shift+Print Screen）保存为“模型编译方法.png”文件。

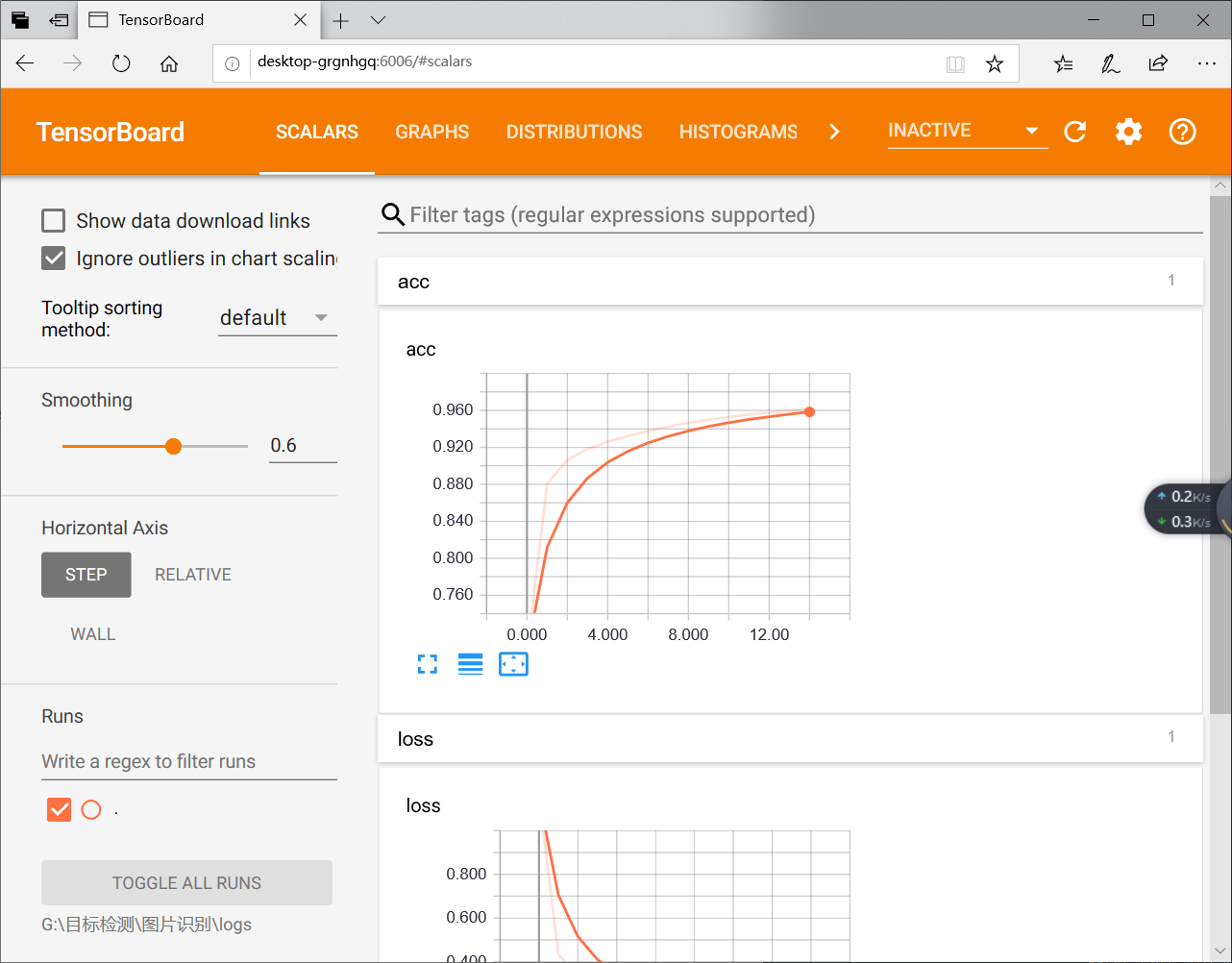
将代码截图文件保存在桌面任务文件夹下。



### 训练过程可视化呈现

TensorBoard方法及参数设置”待训练完毕后，在pychram的terminal中使用相关命令加载TensorBoard日志文件将训练过程可视化展现在浏览器中，将浏览器的TensorBoard可视化页面截图（shift+Print Screen）保存为“训练过程可视化.png”文件。



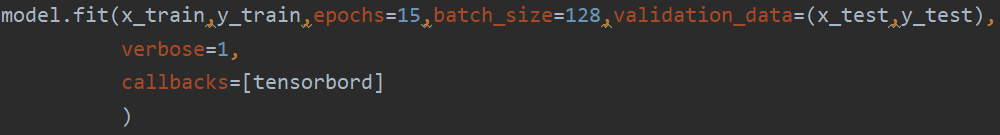


将训练过程可视化截图文件保存在桌面任务文件夹下。

### 完成模型训练方法

完成模型训练方法及参数设置”，并截图（shift+Print Screen）保存为“模型训练方法.png”文件；并截图（shift+Print Screen）保存为“评估模型方法.png”文件。

将代码截图文件保存在桌面任务文件夹下。



### 保存模型文件

完成保存模型方法”编写代码将模型保存为“model.h5”文件文件进行训练，待训练完成后将控制台中打印出的模型评估结果截图（shift+Print Screen）保存为“模型评估结果.png”文件。

将模型评估结果截图文件及生成的模型文件保存在桌面任务文件夹下。



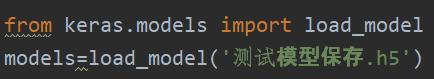
## 任务四 模型加载及预测

经过上一个任务，我们已经训练好了一个可以对这十种菜品进行识别的人工智能模型，本任务将使用模型进行模型的加载及预测工作，。完成本任务需要的全部资源列表如下：

### 完成模型加载方法

完成模型加载方法”，并截图（shift+Print Screen）保存为“模型加载方法.png”文件。

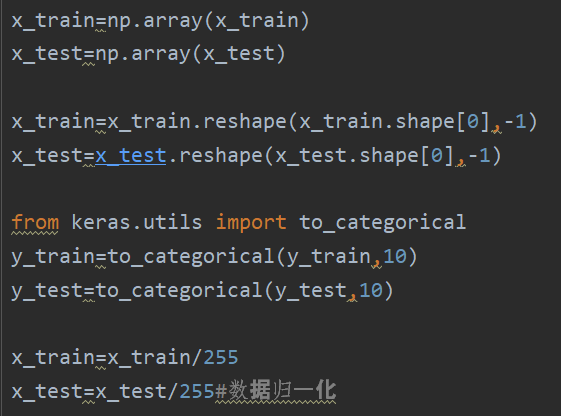
将代码截图文件保存在桌面任务文件夹下。



### 完成待预测图片的预处理

完成待预测图像预处理”，并截图（shift+Print Screen）保存为“待预测图像预处理.png”文件。

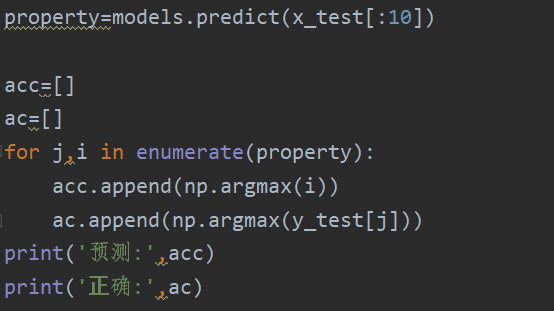
将代码截图文件保存在桌面任务文件夹下。



### 完成模型预测方法

完成模型预测方法”按照注释及其括号中的详细说明编写代码，并截图（shift+Print Screen）保存为“模型预测方法.png”文件。

将代码截图文件保存在桌面任务文件夹下。



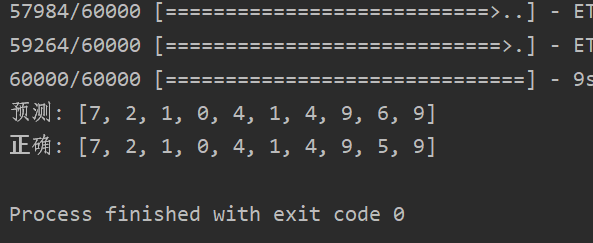
### 4.对待预测图片生成预测结果

完成模型预测方法步骤后运行model\_application\_predict.py，在Pycharm的控制台中对两张待预测图像分别进行预测并生成预测结果，将控制台中的预测结果截图（shift+Print Screen）保存为“模型预测结果.png”文件。

将预测结果截图文件保存在桌面任务文件夹下。

1. 综上完成web应用的开发

完成用户在web客户端上传一张图片，后端实现对此图片进行预测并生成预测结果返回至客户端显示，保存html代码截图、flask路由配置截图、



## 任务五 提交项目综合分析报告

经过上述一系列操作，我们已经明确了训练一个人工智能模型的基本流程，从以下几个方面对本次竞赛任务进行分析，撰写综合分析报告。完成本任务需要的全部资源列表如下：

1. 总结运用TensorFlow训练一个人工智能模型需要进行哪些步骤。

加载数据集，清洗数据集，对模型进行构造编译，保存模型，读取模型，预测数据

1. 深度学习和传统的机器学习有哪些不同的地方。
2. 更6
3. 假如需要对这个模型的效果进行改进，可以从哪些方面入手。

训练的次数和替换更适合的优化器

温馨提示：所有任务均需提供运行效果截图、以及代码截图

以及所有代码命名规范提交

任务以task.x.x.py命名，其他提交文件类似。

五个任务分五个文件夹依次task1、task2....存入

**最后提交报告有两个，一个综合分析报告，一个答题思路报告（运行效果截图、代码截图）**