## 深度学习平台及框架调研报告

# 1、华为ModelArts

**1.1、平台简介：**ModelArts 支持海量数据预处理、大规模分布式训练、自动学习、具备端-边-云模型按需部署能力，可帮助用户快速训练和部署模型，管理全周期 AI 工作流。



使用过程示例：<https://baijiahao.baidu.com/s?id=1626599147172030990&wfr=spider&for=pc>

**1.2、支持框架：**Tensorflow、MXNet、Caffe、PyTorch

**1.3、业务开发者-自动学习：**

**此类开发者只需进行数据标注，平台已提供基本的业务类型。**

自动学习包括图像分类、物体检测、预测分析、声音分类四个部分

同时数据集标注部分详见【AI初学者-使用预置算法开发】中数据集标注模块。

**1.4、AI初学者-使用预置算法开发：**

**此类开发者只需进行数据标注，并选择合适的预置算法。**

**1.4.1、数据集标注：**

目前数据集标注仅支持“图像分类”、“物体检测”、“文本分类”、“声音分类”四种类型的数据集；

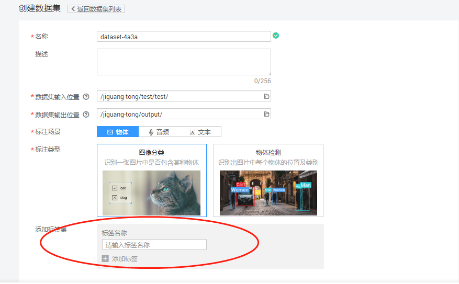
图像分类标注：

第一步：将图片数据上传到华为对象存储服务OBS中；

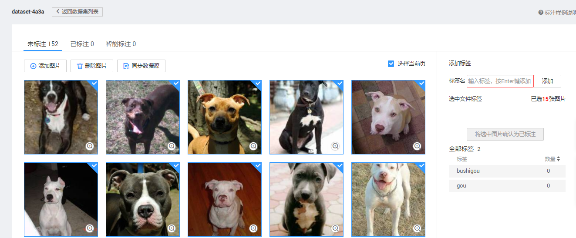
第二步：到ModelArts界面中点击数据标注模块，创建数据集；



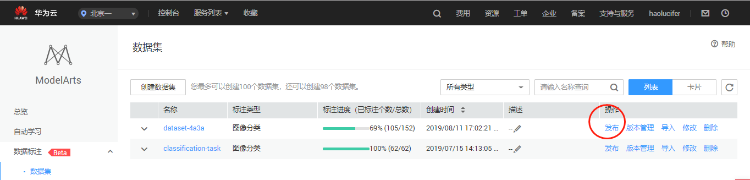
第三步：选择创建数据集类型，根据提示进行创建，圆圈部分为图像分类中的标签；



第四步：设置完标签后，界面就会出现如下标记界面，左边选择图片，右边原则对应的标签，可批量选择；



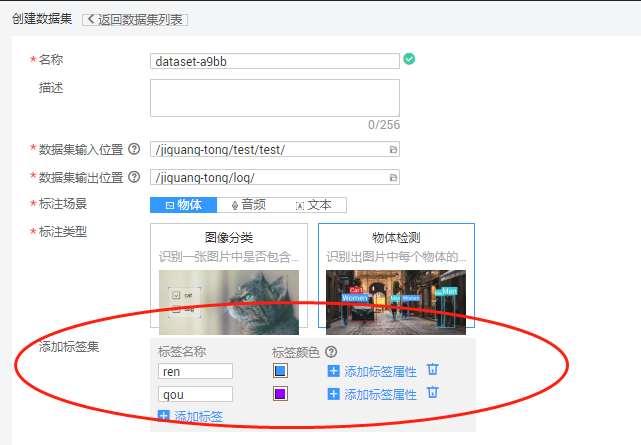
第五步：全部标注完，即返回上层，点击“发布”即可：



图像分类的第二种标注法：将相同标签的图片放在一个目录里，并且目录名字即为标签名。只支持JPG\JEPG\PNG\BMP格式的图片；

目标检测：目标检测的简易模式要求用户将标注对象和标注文件存储在同一目录，并且一一对应，如标注对象文件名为‘img\_20190715.jpg’，那么标注文件的文件名应为‘img\_20190715.xml’,物体检测的标注文件需要满足PASCAL VOC格式；

目标检测的标签标记方法如图像分类的一样，不同的是上述第三步，如下图所示，标签有相应的颜色属性，同时还可以自己添加特有属性：



目标检测的标记需要在原图上进行画矩形框，因此，画法也需要特别注意，如下图所示：



文本分类：文本分类的标注对象和标注文件均为文本文件，并且以行数进行对应，如标注文件中的第一行表示的是标注对象文件中的第一行的标注。

声音分类：声音分类的简易模式要求用户将相同标签的声音文件放在一个目录里，并且目录名字即为标签名，类似于图像分类。

**四种类型的数据集可以做如下七种数据标注：**

**图像分类：识别一张图片中是否包含某种物体**

**物体检测：识别出图片中每个物体的位置及类别**

**声音分类：对声音进行分类**

**语音内容：对语音内容进行标注**

**语音分割：对语音进行分段标注**

**文本分类：对文本的内容按照标签进行分类处理**

**命名实体：针对文本中的实体片段进行标注，如“时间”、“地点”等**

**1.4.2、算法种类（物体检测的训练数据集对应着如上数据标注的数据集，图像分类同上，图像语义分割则没有提供数据标注的入口）：**

**预置算法简介（训练参数描述，无网络细节描述）：**

<https://support.huaweicloud.com/beginners-modelarts/modelarts_22_0034.html>

**MXNet-1.2.1-python2.7:**

yolo\_v3(物体检测)

darknet\_53(图像分类)

SegNet\_VGG\_BN\_16(图像语义分割)

ResNet\_v2\_50(图像分类)

Faster\_RCNN\_ResNet\_v2\_50(物体检测)

**TF-1.8.0-python2.7:**

retinanet\_resnet\_v1\_50(物体检测)

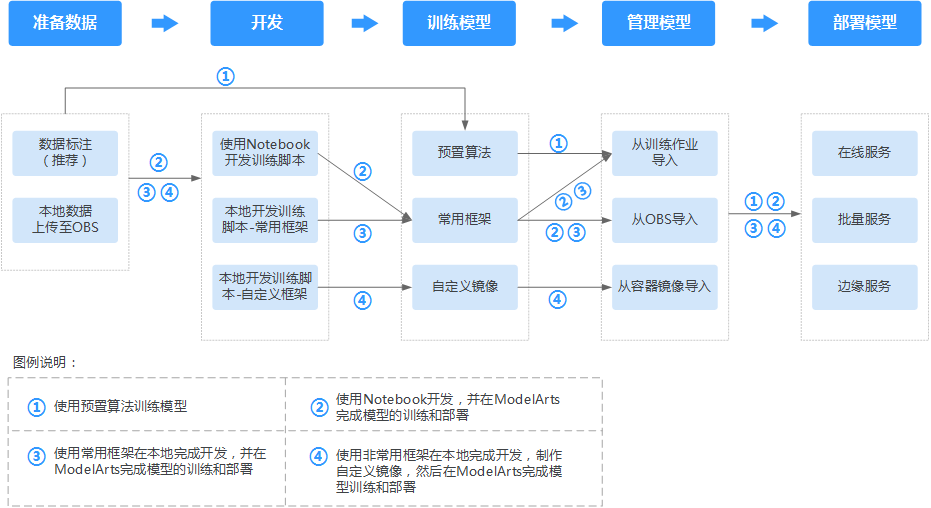
inception\_3(图像分类)

ResNet\_v1\_50(图像分类)

Faster\_RCNN\_ResNet\_v1\_50(物体检测)

**1.5、AI工程师-全流程开发：**

**此类开发者可自由选择开发难度，下图为此类开发者开发流程：**



# 2、百度EasyDL定制化训练平台

此平台类似于华为ModelArts中的1.3：业务开发者-自动学习，他不提供可供选择的通用内置算法。

平台应用步骤包括如下步骤：

Step 0: 选择模型（图像分类、物体检测、声音分类、文本分类、视频分类）

Step 1：创建模型（模型名字、联系方式、功能描述）

Step 2：上传并标注数据（创建数据集模块制作好模型对应的数据集即可）

Step 3：训练模型（如图像分类模块中的图所示，选择创建的模型，选择算法【此算法不开源名字，统称通用算法】）

Step 4: 校验模型效果

Step 5：发布模型-在线API

**图像分类：**识别一张图中是否是某类物体/状态/场景。可以识别图片中主体单一的场景

数据集上传方法：zip格式的文件，下图展示训练可供选择的算法界面：



（注：以下所有场景均无法看到算法名字，在此不做坠余描述）

**物体检测：**在一张图包含多个物体的情况下，定制识别出每个物体的位置、数量、名称。可以识别图片中有多个主体的场景

**声音分类：**识别出当前音频是哪种声音，或者是什么状态/场景的声音

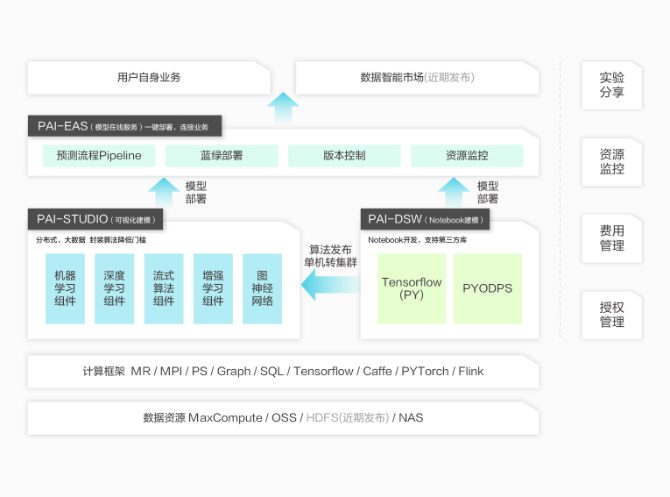
**文本分类：**基于自建分类体系的机器学习方法，可实现文本自动分类

**视频分类：**分析短视频的内容，识别出视频内人体行为动作，环境变化，或是物体位置/状态变化

（附：数据集标注大体跟ModelArts一致，由于需要费用，无法截图说明详情）

# 3、阿里机器学习Pai

**PAI的组成部分包括三个部分：**PAI Studio可视化建模，PAI DSW Notebook建模，PAI EAS模型在线服务。三者的关系如下图所示：



**3.1、PAI Studio:**

**3.1.1、支持框架：**Tensorflow(tf1.4、tf1.8)-案例（手写数字识别）、Caffe rc3、MXNet 0.9.5-案例（CIFAR-10数据集自动分类）, Tensorflow 和 MXNet 支持用户自己编写的python代码，Caffe支持用户自定义网络文件。

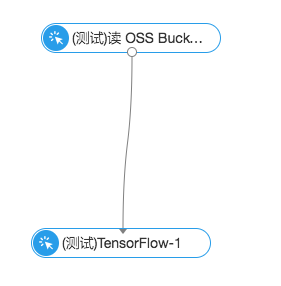
**3.1.2、简介：**为开发者提供可视化的机器学习实验开发环境，帮助用户实现无代码开发人工智能相关服务。在PAI Studio中，可引用首页已有的模板实验，完成创建后可直接运行部署。创建自定义实验时，平台内置了百余种组件化算法供选择，同时支持接入多种数据源包括MaxCompute表数据以及OSS数据。运行实验进行模型训练时，支持AutoML自动调参辅助用户找到最佳模型。模型支持导出PMML以及模型在线部署，对外提供API服务。

此平台的特点是可视化界面，通过拖拽图标的形式完成训练操作，如下演示用tensorflow（MXNet\Caffe的过程是一样的）训练mnist数据集的过程 (OSS是阿里平台的对象存储服务)：

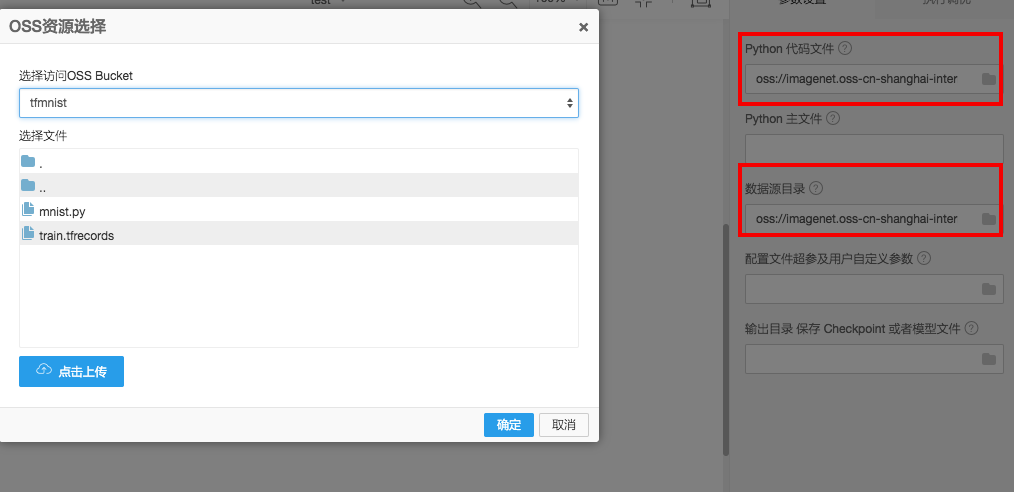
1、在OSS端上传python执行文件以及训练数据集。本案例在OSS华东2 region创建bucket，bucket名为tfmnist，上传python脚本以及训练数据。



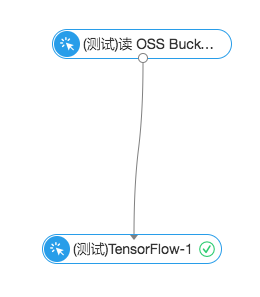
2、拖拽读OSS Bucket和TensorFlow组件，拼接成如下实验。需要设置好OSS Bucket的地区，并且完成RAM授权，如下图所示。



3、配置TensorFlow组件参数。参照下图配置python执行文件以及数据源文件路径，如下图所示。



4、单击运行，直到两个组件运行完成，如下图所示（右下角有绿色的勾号）。



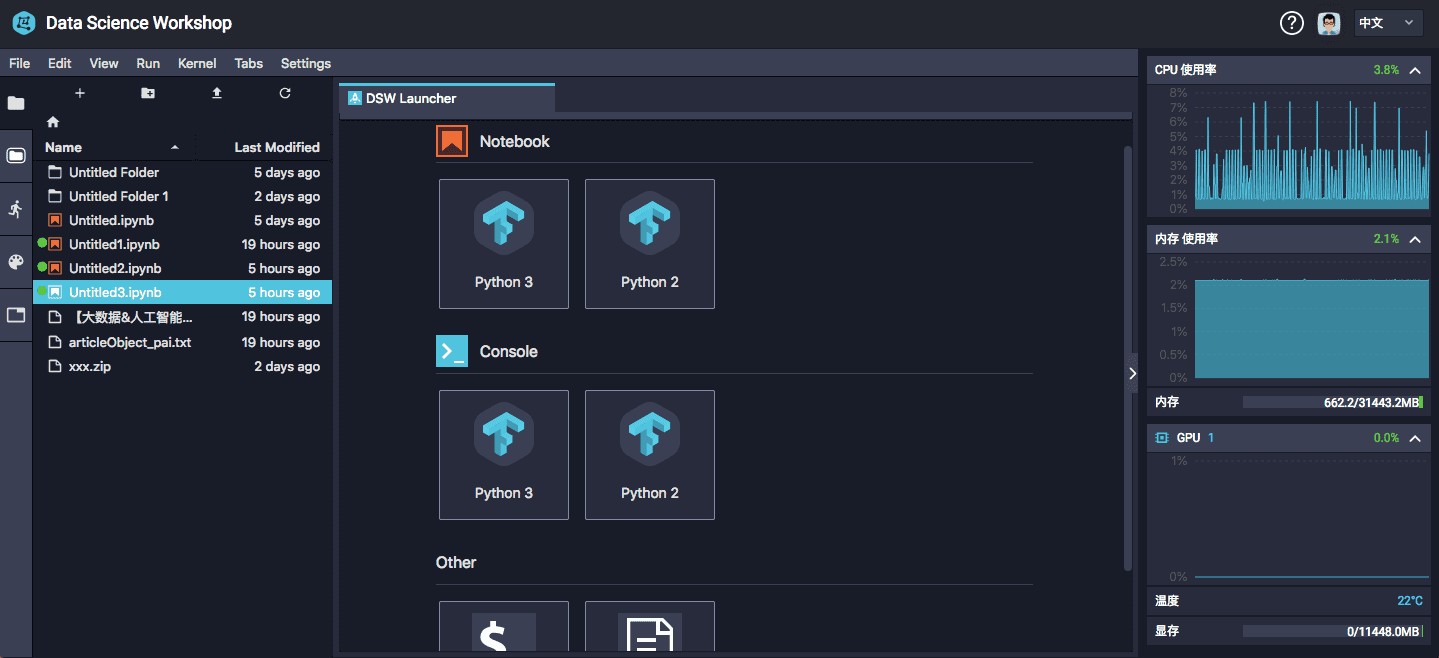
5、右键单击TensorFlow组件，查看运行日志。



**3.2、PAI DSW Notebook**

**3.2.1、支持框架：Tensorflow**

**3.2.2、简介：**PAI-DSW（Data science workshop）是专门为算法开发者准备的云端深度学习开发环境，用户可以登录DSW进行代码的开发并运行工作。目前DSW内置了PAI团队深度优化过的Tensorflow框架，同时也可以通过打开console对话窗口自行安装需要的第三方库。操作界面如下图：



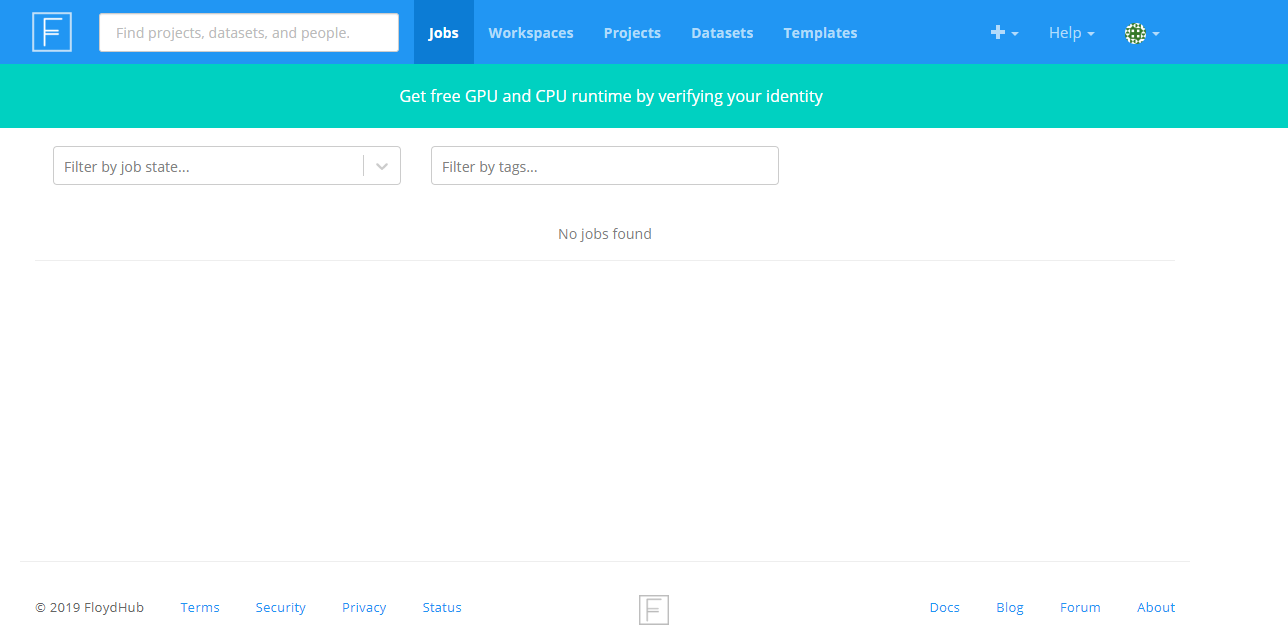
**3.3、PAI EAS**

**3.3.1、简介：**在线推理是将算法模型应用至实际业务的重要环节。为了帮助用户更好的实现一站式端到端的算法应用，PAI平台针对在线推理场景提供了PAI EAS(Elastic Algorithm Service)在线预测服务，支持基于异构硬件(CPU/GPU)的模型加载和数据请求的实时响应。您可以通过多种部署方式将您的模型发布成为在线的restful API接口，同时我们提供的弹性扩缩、蓝绿部署等特性可以支撑您以最低的资源成本获取高并发、稳定的在线算法模型服务。

# 4、Floyd

**4.1、支持框架：**Pytorch、Tensorflow

**4.2、简介：**此平台是集成计算资源为主的一个平台，他的主目录中有（任务（jobs），工作空间(workspaces)，项目(project)，数据集(datasets)，模板(templates)）这几项，初始界面如下：



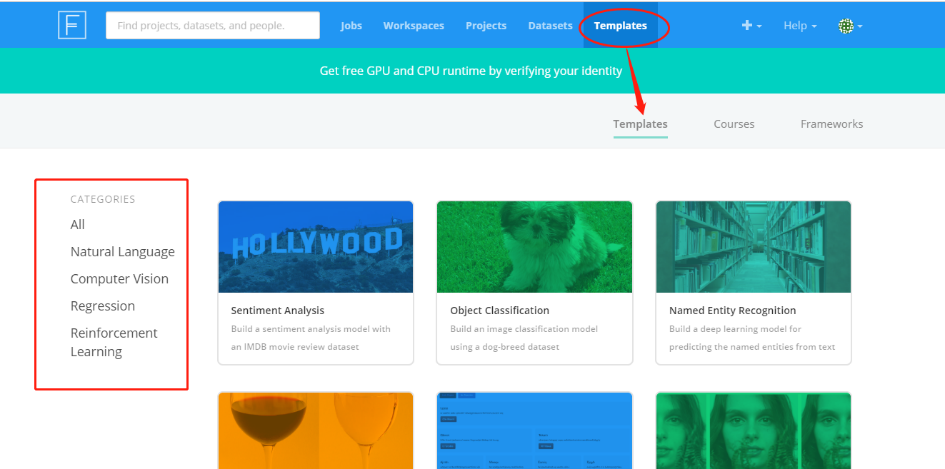
其中，此平台是不提供制作数据集标签的服务的，也即没有通用算法的封装。

**4.3、特色**

该平台的模板（Templates）专区有三个模块分别为（Templates, Courses, Frameworks）。

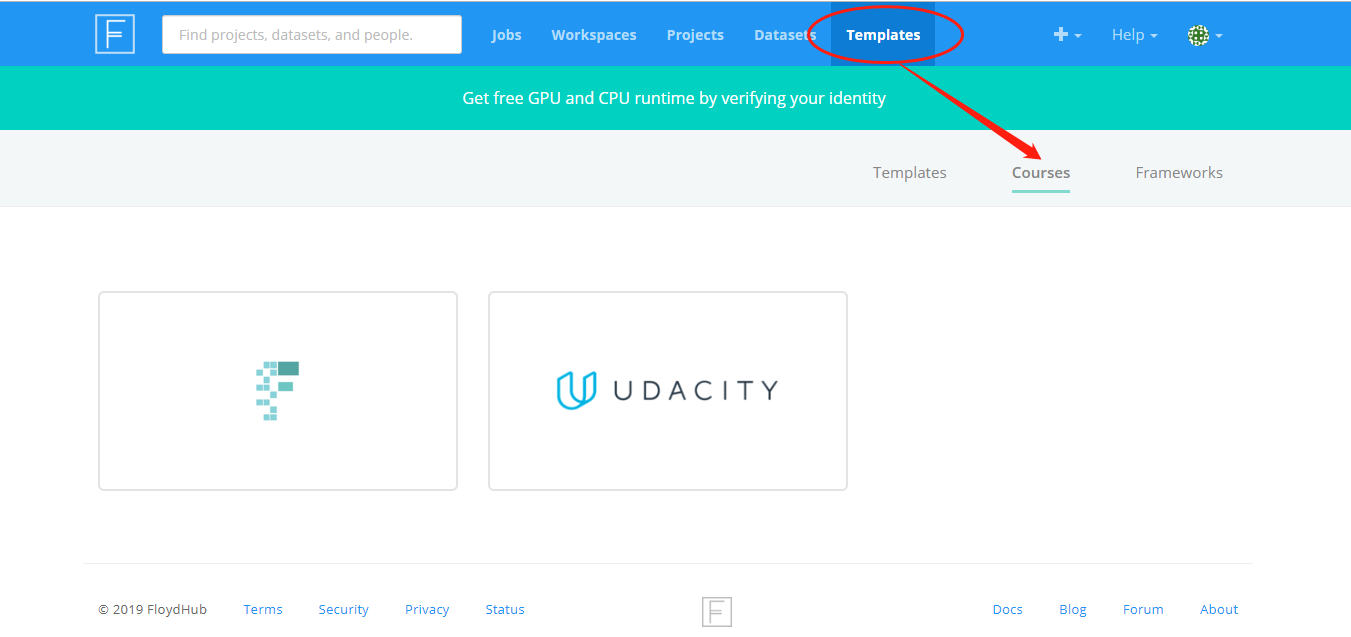
**4.3.1、Templates**

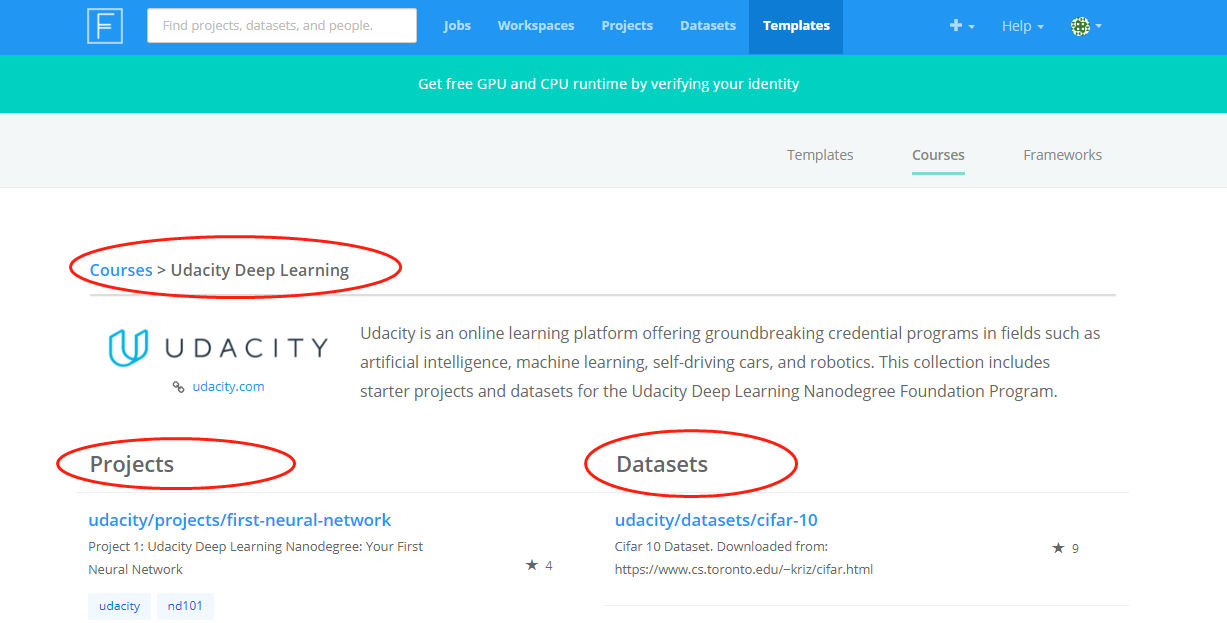
如下图所示，模板这一栏集成了各领域（自然语言处理、计算机视觉、回归任务、强化学习）的通用模板，用户可以根据需要使用模板，用户一旦使用了模板，即为在自己的账户下创建了此项目。



**4.3.2、Courses**

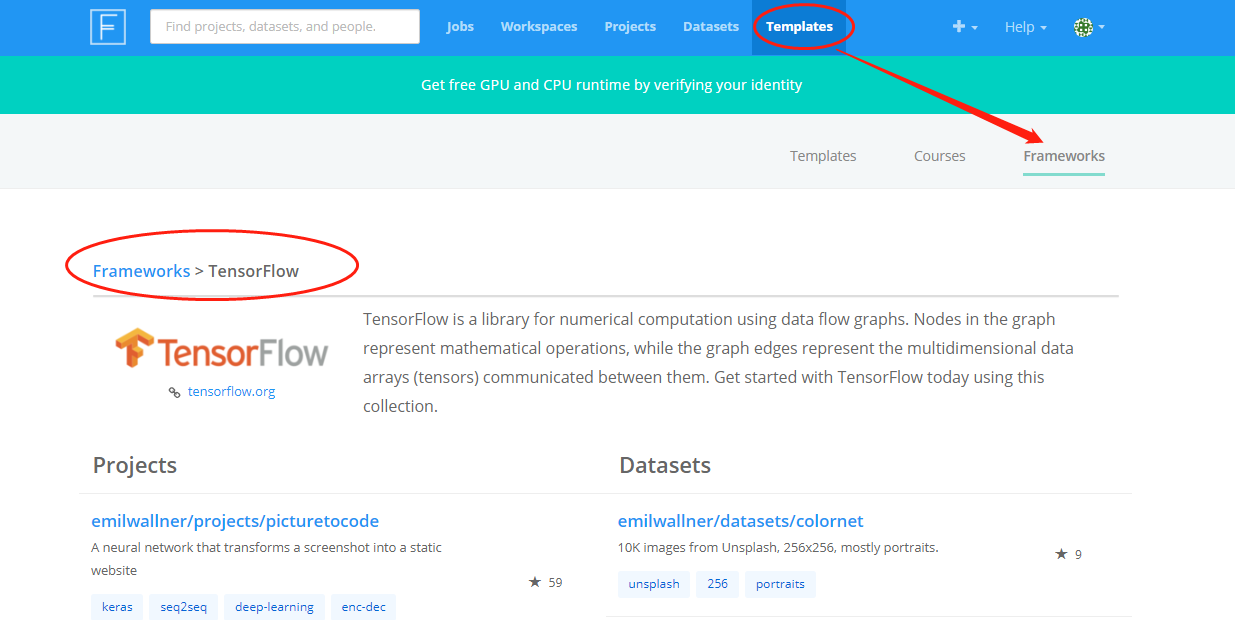
教程模块主要提供了一些如何利用floyd平台创建项目和下载公用的数据集，方便大家快速入门平台的使用，界面截图如下：





**4.3.3、Frameworks**

框架模块公开了基于tensorflow和pytorch框架的一些算法，主要也是供大家学习使用。



# 5、AIDP

**5.1、简介**：AIDP（artificial intelligence development platform)是中国移动苏州研发中心（苏小研）的深度学习私有云平台，集成了包括tensorflow、pytorch、mnist、caffe等主流深度学习框架，使用户可以不需要关心框架本身的部署，从而方便用户进行深度学习研究。

**5.2、与公有云的区别：**私有云不同于公有云，首先AIDP只有在苏小研公司局域网内部才可以使用；其次，目前，AIDP的存储设计还没有做到跟各大公有云平台那样，数据跟算法隔离；另外，AIDP目前仅集成了深度学习框架，还没有类似于其他平台那样，内置一些通用的深度学习算法。

## 各框架介绍

# Tensorflow

**TensorFlow 的优点**

（1）安装便捷；

（2）巨大且活跃的社区，可以在学习过程中少走弯路；

（3）功能强大，不仅支持深度学习，还支持强化学习或其他机器学习算法。

**TensorFlow 的缺点**

（1）新概念较多，类似于palceholder、operation、session等概念需要理解；

（2）底层接口写起来繁琐，高层接口又不灵活，而且它的高层封装不好实现；

（3）它主要面向超大数据量的训练，因此计算速度相对较慢；

（4）各个版本之间的兼容性较差，需要进一步完善；

（5）相对于pytorch较不灵活，多用于工业界。

# MXNet

**MXNet 的优点**

（1）相当快的评测结果;

（2）彻底的灵活性;

（3）支持多 GPU 训练;

（4）支持多种设备。

**MXNet 的缺点**

（1）学习社区较小，学习文档较少;

（3）安装调试较麻烦。

# Pytorch

**Pytorch的优点**

（1）简洁易懂，可读性很强；

（2）运行速度快；

（3）有大量预训练好的模型，易于组合的模块化组件；

（4）活跃的社区;

（5）较tensorflow更灵活，多用于学术界。

# Keras（基于tensorflow）

**Keras的优点**

（1）简易和快速的原型设计；

（2）活跃的社区；

（3）它是更高级，对用户最友好的API，易于阅读和使用，适用于初学者；

**Keras 的缺点**

（1）不支持 seq2seq，因此实现 NLP 比较困难；

（2） 在 TensorFlow backend 时， 比TensorFlow 慢；

（3）没有增强学习工具箱，自己修改实现很麻烦；

（4）封装得较高级，训练细节不能修改，不合适算法研究。

# Caffe

**Caffe 的优点**

（1）专精于图像处理；

（2）其Model Zoo中有大量训练好的经典模型；

（3）稳健且快速，适用于工业应用；

**Caffe 的缺点**

（1）安装调试比较繁琐困难，有大量的依赖包；

（2）各个版本之间的兼容性较差；

（3）学习文档较少；

（4）不够灵活。

**五大深度学习框架对比**

**框架**

**对比项**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Tensorflow** | **Pytorch** | **MXNet** | **Caffe** | **Keras** |
| 主语言 | C++/cuda | python | C++/cuda | C++/cuda | python |
| 接口 | python/matlab/java/Go.. | python | python/R/Julia/Go.. | python/matlab/protobuf | python |
| 硬件 | CPU/GPU/Mobile | CPU/GPU/Mobile | CPU/GPU/Mobile | CPU/GPU | CPU/GPU |
| 分布式 | Y | Y | Y | N | Y |
| 速度 | 中等 | 中等 | 快 | 快 | 慢 |
| 灵活性 | 较好 | 好 | 较好 | 一般 | 一般 |
| 文档 | 全面(中/英) | 全面(中/英) | 全面(英) | 一般（英） | 全面(中/英) |
| 适合  模型 | CNN/RNN | CNN/RNN | CNN | CNN | CNN/RNN |
| 命令式 | N | Y | Y | N | N |
| 声明式 | Y | N | Y | Y | Y |
| 社区 | 活跃 | 活跃 | 一般 | 一般 | 一般 |
| 开发者 | Google | FaceBook | 李沐/陈天奇等 | 贾杨清 | Google |