UCLOUD

悟有所值 Ucan下午茶 Ucan Afternoon Tea

AI 公有云实践

范融 UCloud深度学习工程师

UCLOUD 专业云计算服务商

IT大咖说 知识共享平台

目录

01

为什么要建设AI公有云平台

AI运用的各类场景

零售

安防

医疗

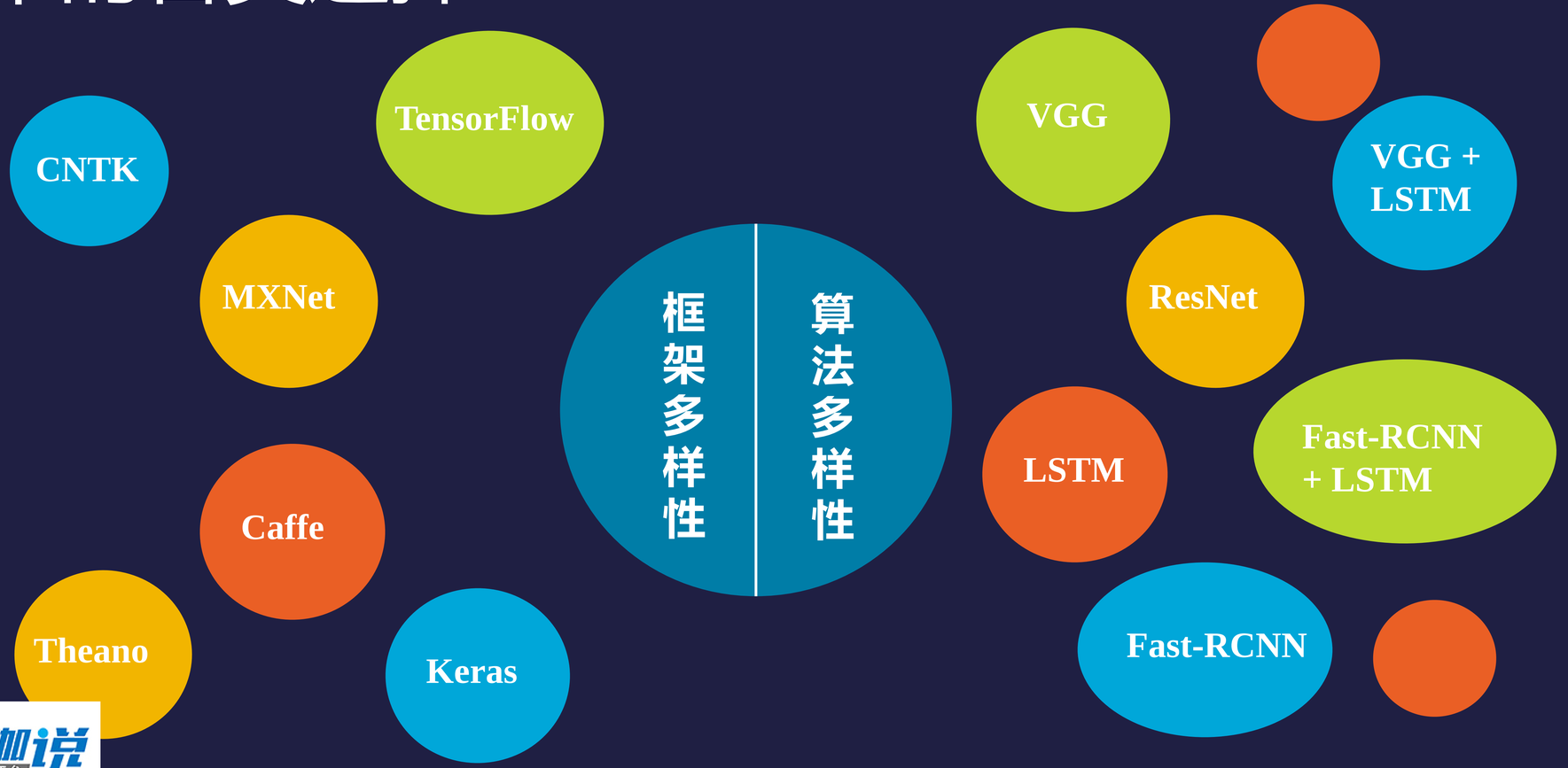
教育

艺术

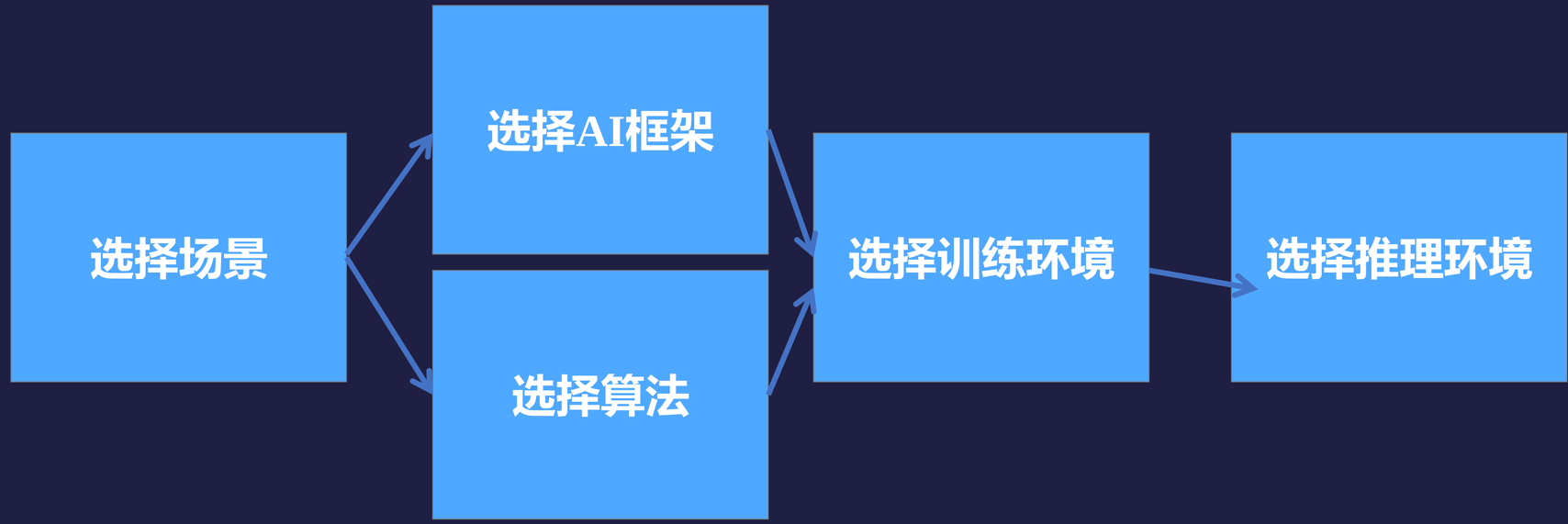
游戏

金融

AI技术的各类选择



AI项目研发周期



AI平台需求（技术平台）

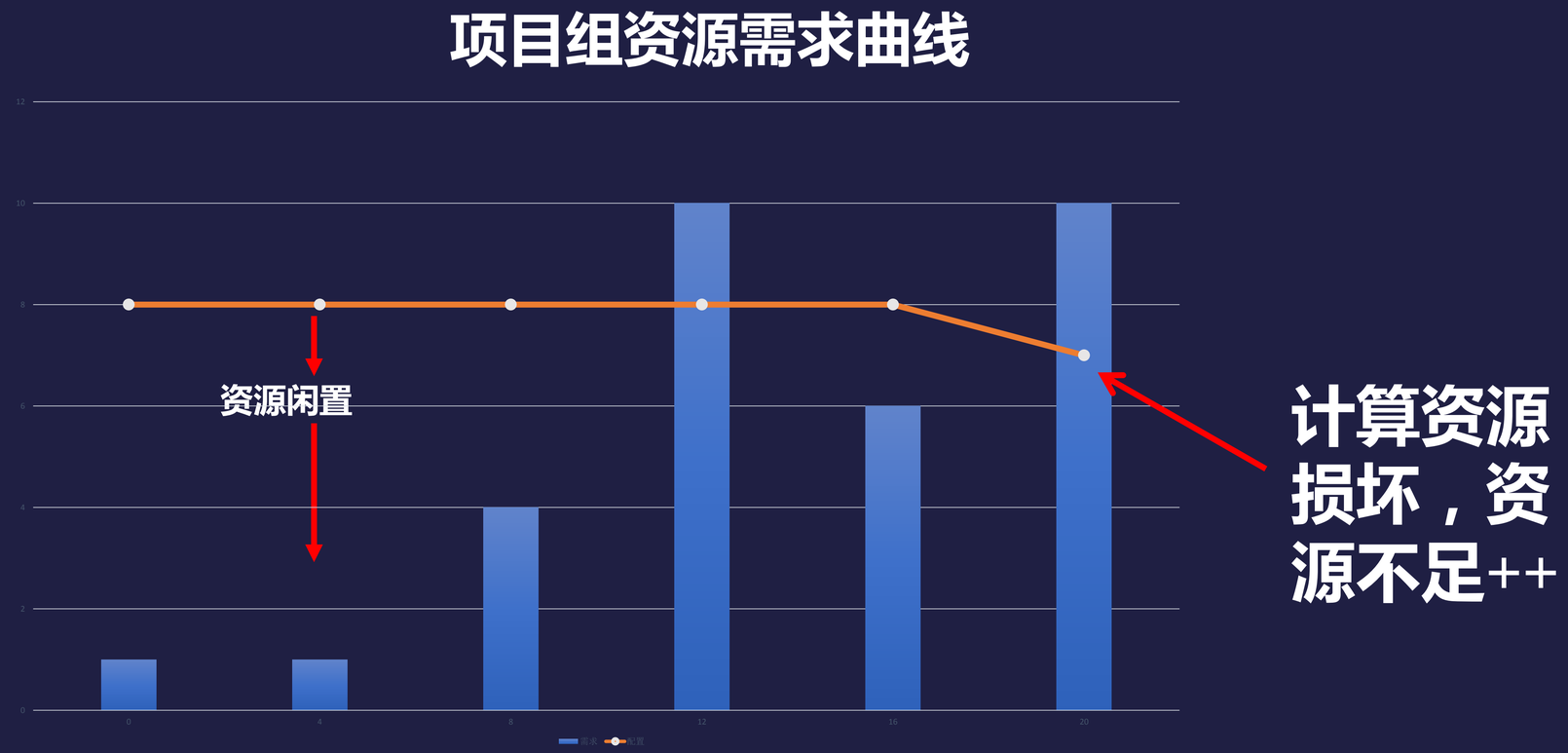
AI 基础平台

计算
CPU、GPU

存储
S3、NSF、HDFS

网络
单节点、分布式

AI平台需求（项目需求）



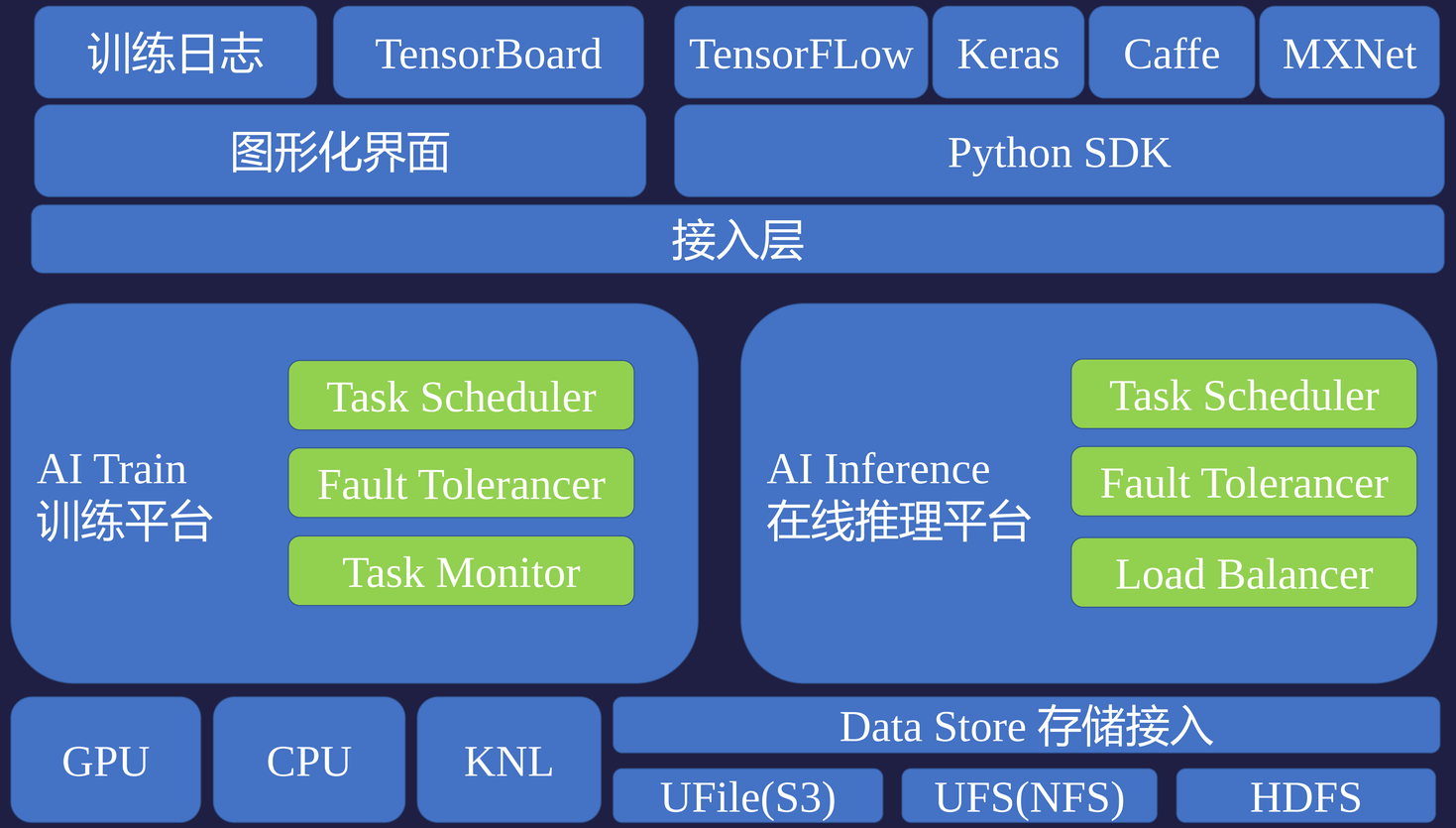
AI平台的兼容性

算法兼容性	更好地兼容各类AI框架和算法
平台扩展性	平台具备横向扩展能力，支持业务规模的不断扩大
分布式化	具备弹性伸缩的能力以及容灾能力
纵向扩展	同时可以支持CPU、GPU等各种计算设备 同时可以支持S3、NFS、HDFS等多种存储类型
易用性	上手简单，方便进行代码迁移

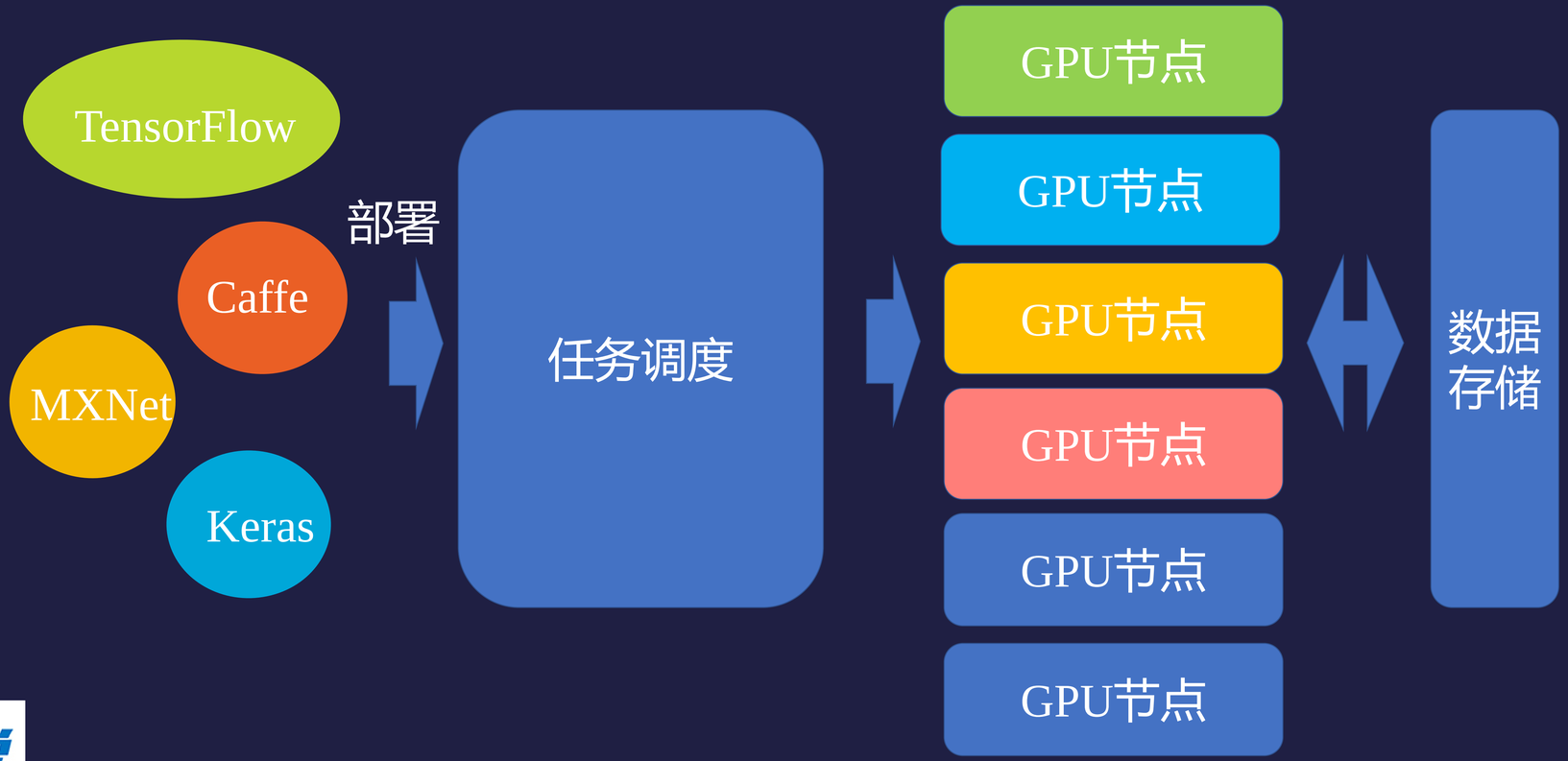
02

UCloud AI基础平台架构

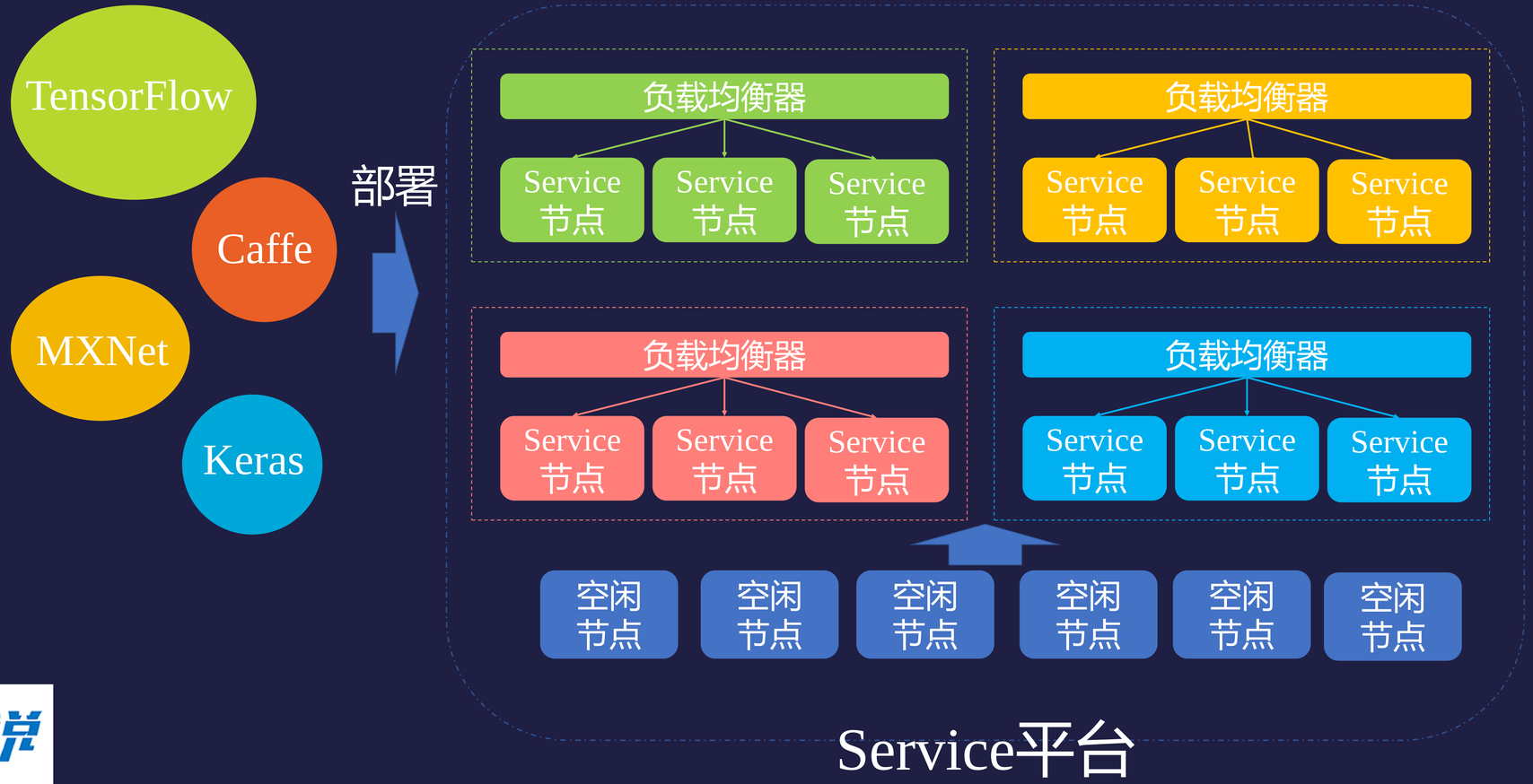
UCloud AI平台整体架构



UCloud AI训练平台



UCloud AI在线推理平台

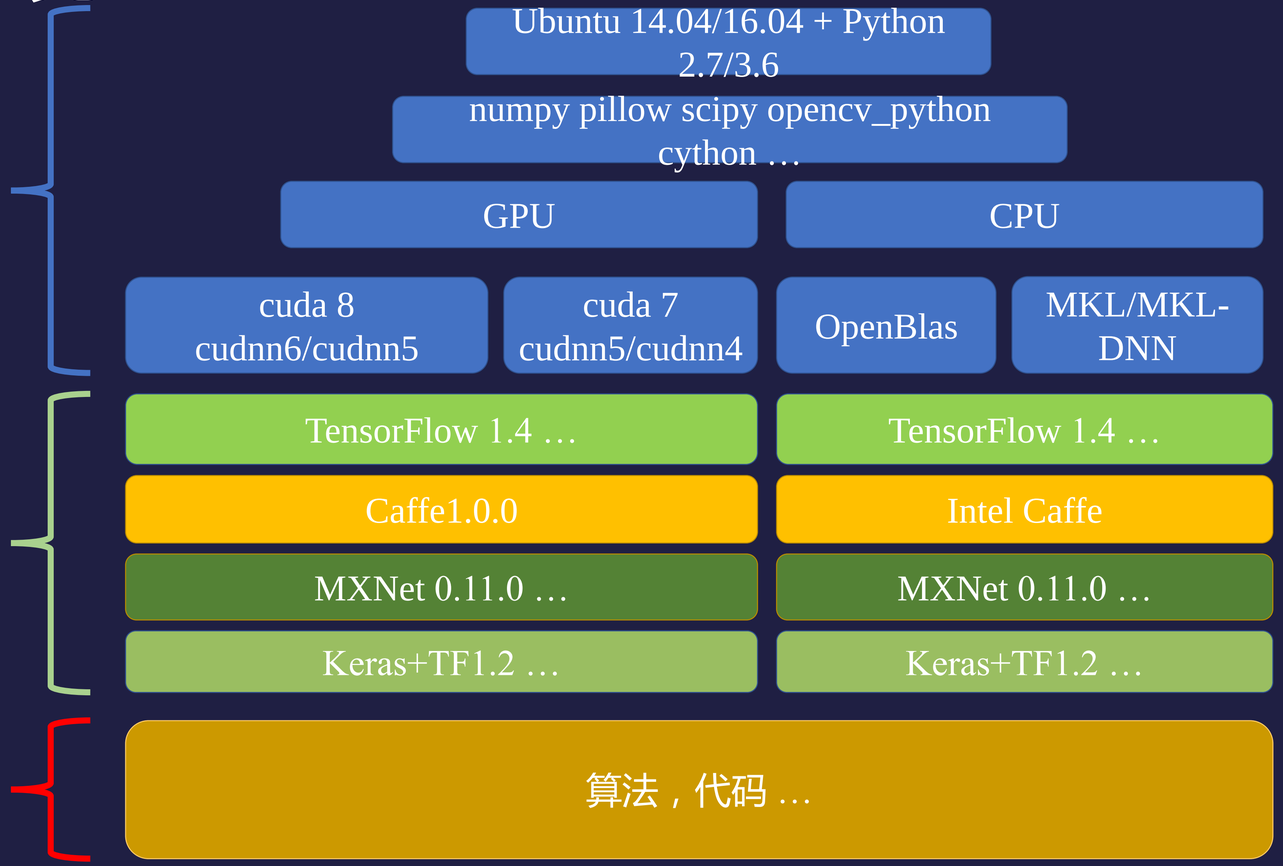


兼容性好的运行环境

统一基础镜像（开源）
- 无需用户考虑
- 自动支持GPU加速

分类基础镜像（开源）
- 用户根据实际使用选择
- 支持各类AI框架

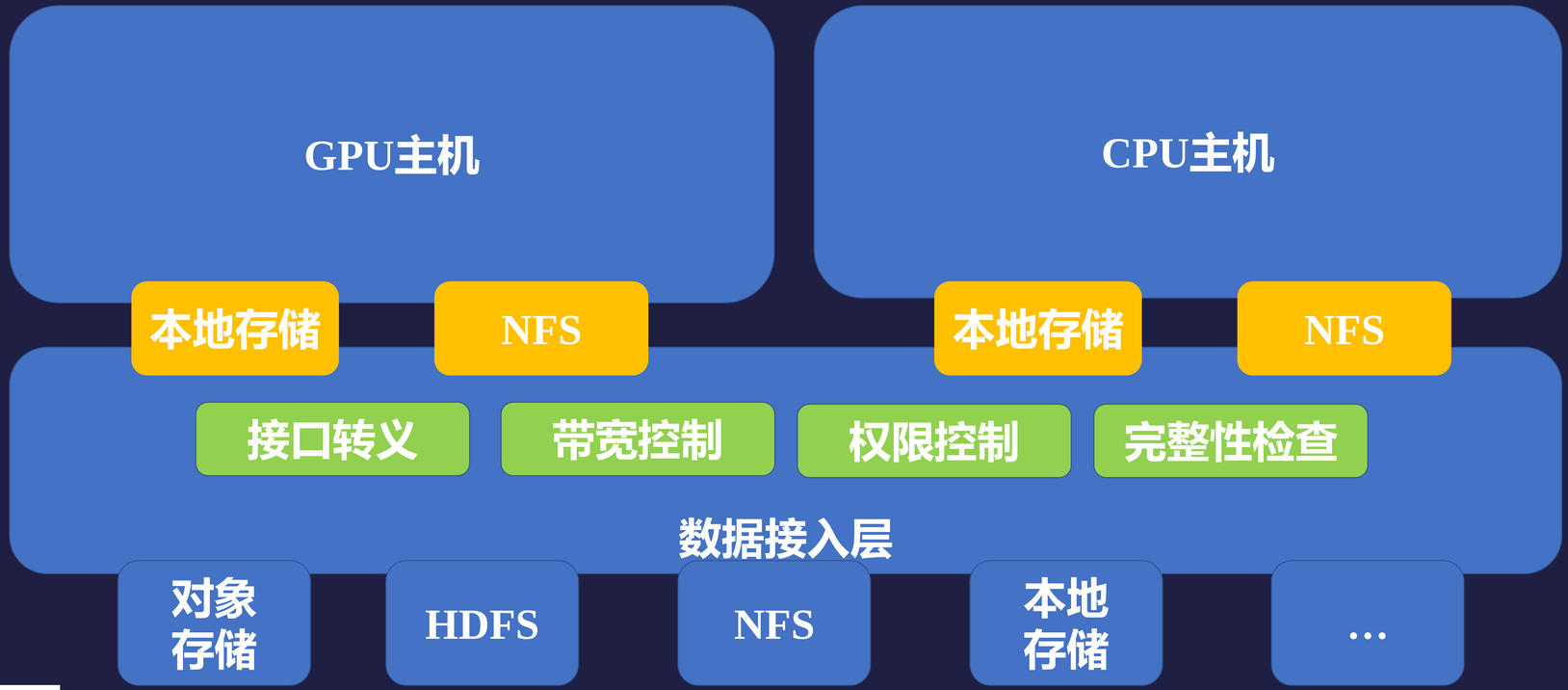
用户自定义镜像
- 用户编写代码



兼容性好的运行环境

封装	运行环境完全隔离，不同任务之间不会产生软件冲突
预装	基础镜像内置各类基础软件环境，减少使用者环境准备开销
自由	可以自由安装各类软件包，封装各类算法
可重用	算法的容器镜像可以重复使用
兼容性	GPU容器镜像可以在任意类型GPU节点运行 CPU容器镜像可以在任意类型CPU节点运行

灵活接入的数据源


GPU主机
CPU主机
本地存储
NFS
本地存储
NFS
接口转义
带宽控制
权限控制
完整性检查
数据接入层
对象存储
HDFS
NFS
本地存储
…

灵活接入的数据源

封装

计算节点逻辑不需要支持各种存储接口，仅需要通过2-3种（例如本地存储、NFS）接口就可以对接各类存储类型

灵活

通过扩展数据接入层可接入的存储类型，也就可以扩展AI平台的数据接入类型

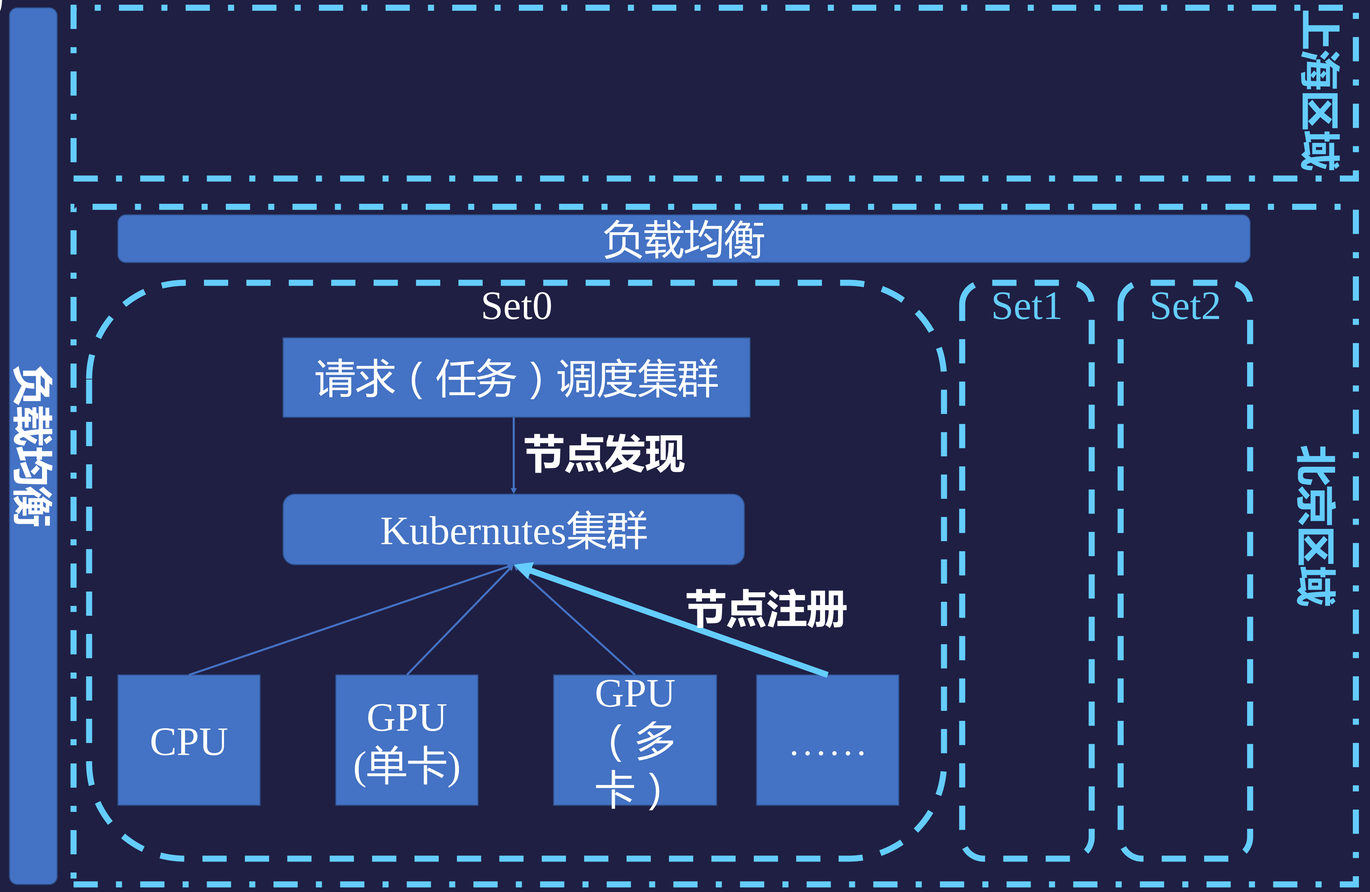
弹性

数据接入层作为中间层可以承载更大的数据访问流量。

安全

数据接入层可以做数据流量控制，确保各个任务的SLA，同时对后端的数据存储系统进行带宽、流量保护

弹性资源调节


上海区域
负载均衡
Set0
Set1
Set2
负载均衡
请求（任务）调度集群
节点发现
Kubernutes集群
节点注册
CPU
GPU
(单卡)
GPU
（多
卡）
……
北京区域

弹性资源调节

兼容 根据训练实际算力需求申请不同机型

弹性 多项目共享资源池，即使申请，即时分配，即时使用
随时扩容缩容，无资源浪费

托管 资源池计算节点损坏自动退出集群，任务恢复，不影响AI业务

03

利用AI平台加速训练实例

利用AI平台加速训练实例

01

图像分类算法简介

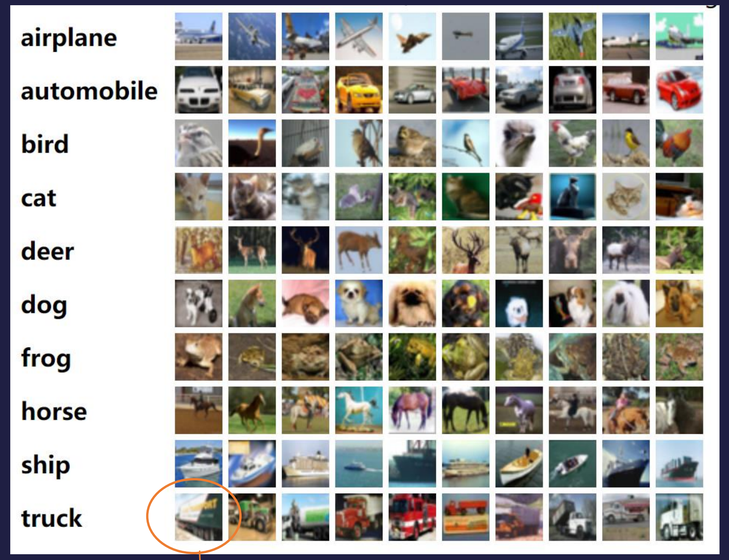
02

图像分类算法云上加速

03

其他开源资源共享

图像分类案例——cifar数据集



大小为32*32
的RGB图像

Cifar10数据集

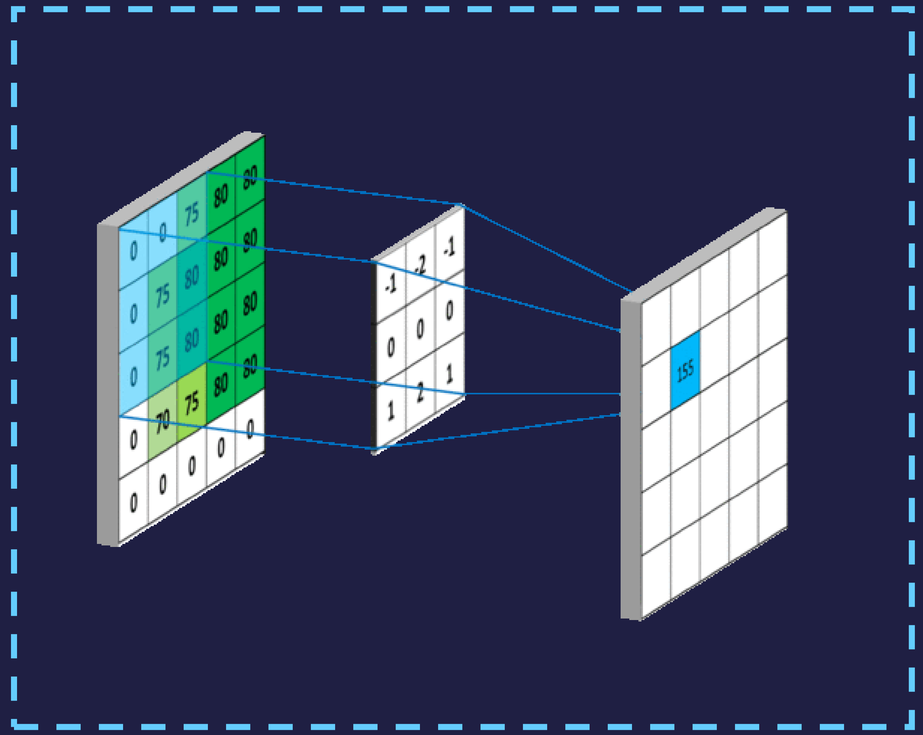
十类物体图片
（10*6000张）
+
相应标签

50000份训练数据

10000份测试数据

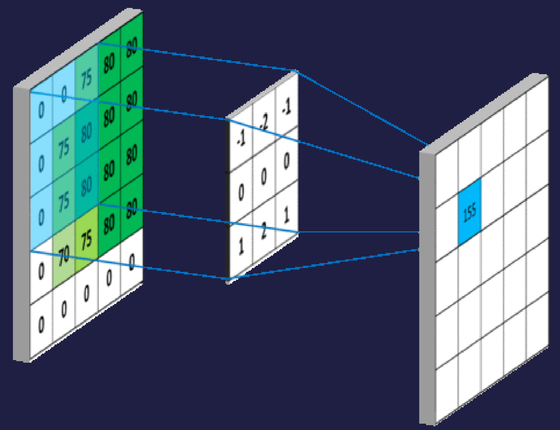
图像分类神经网络

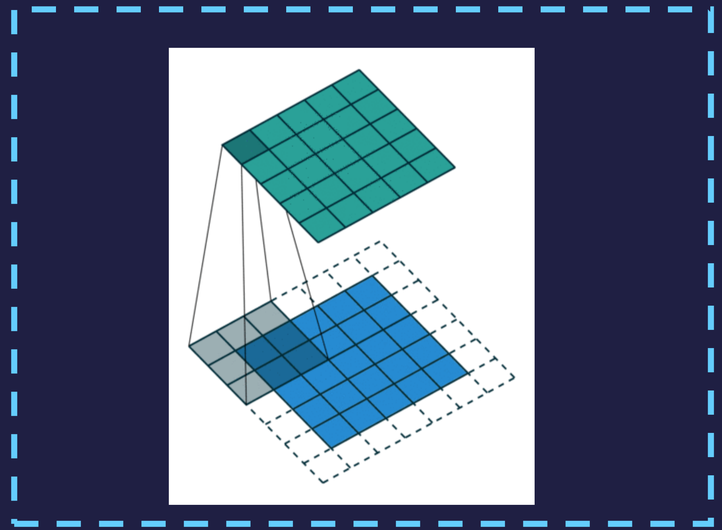
- 卷积计算



图像分类神经网络

- 卷积计算





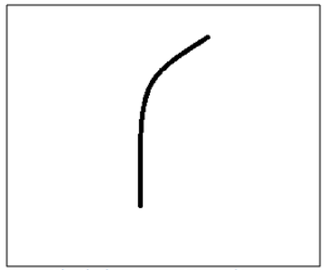
图像分类神经网络

- 卷积计算

卷积核

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

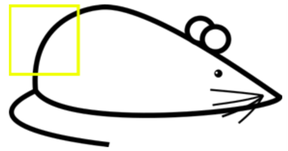
Pixel representation of filter



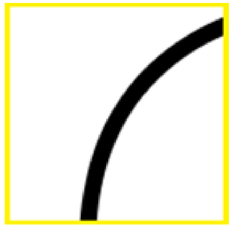
Visualization of a curve detector filter

需要卷积的图像

Original image



Visualization of the filter on the image



Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

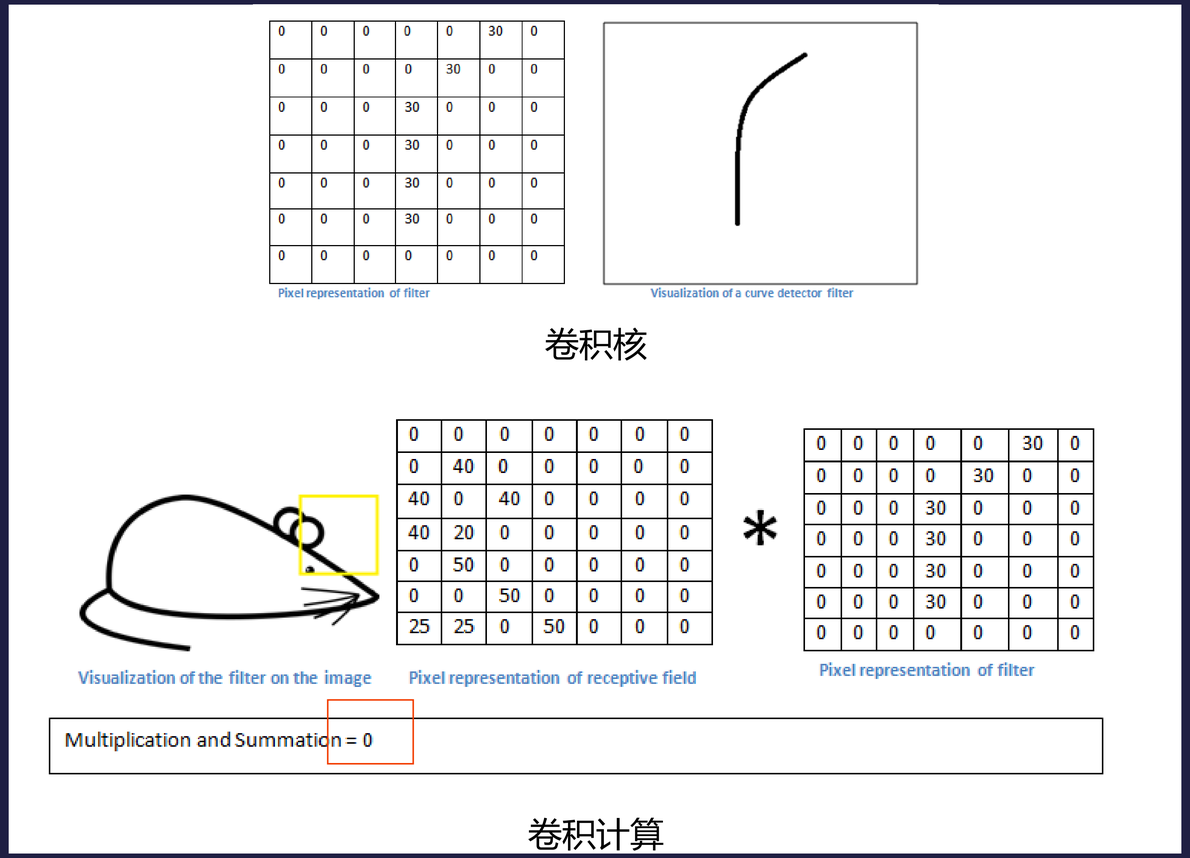
Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

图像分类神经网络

- 卷积计算

卷积运算

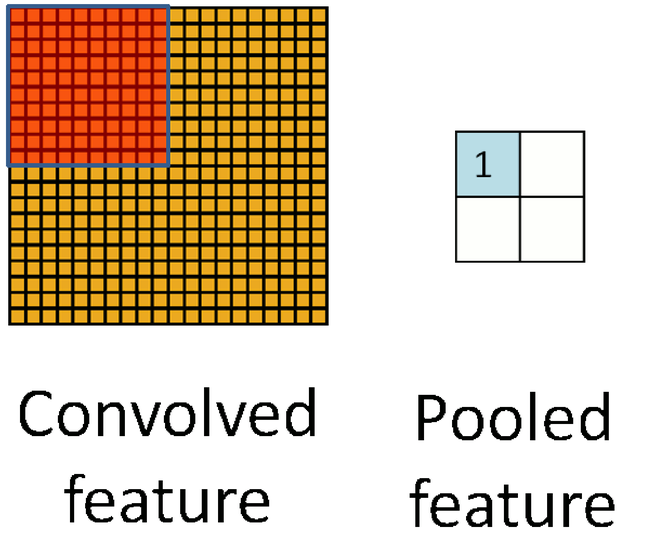
- 提取图像中的特征；
- 不同的卷积核提取不同的特征；
- 训练卷积神经网络，实际上也就是训练每一个卷积层的卷积核，让这些卷积核对特定的模式有高的激活，以达到CNN网络的分类/检测等目的。



图像分类神经网络

- 池化计算

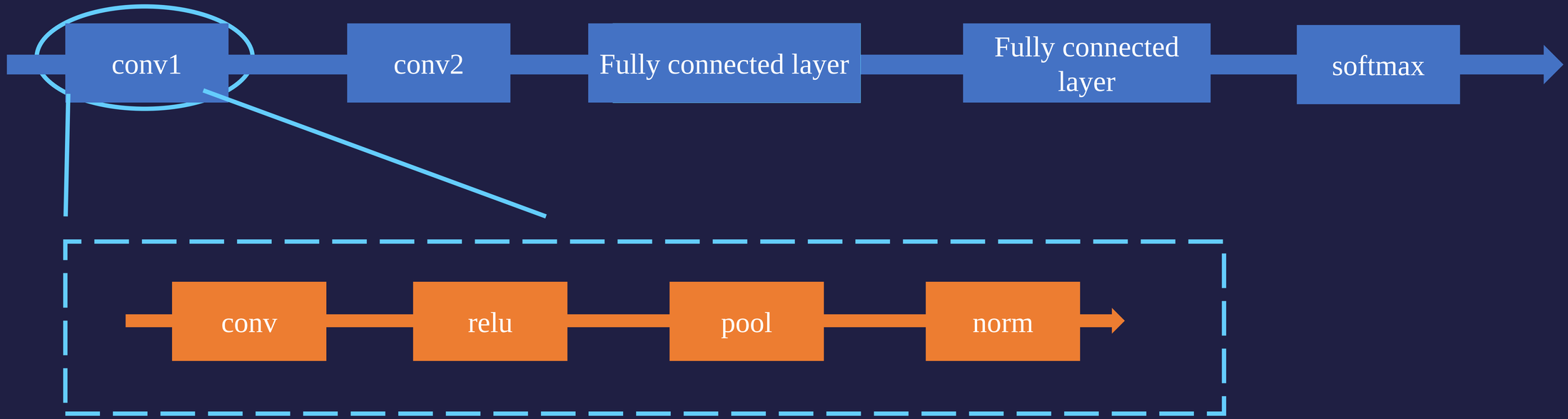
池化运算
提取主要特征；
减小计算复杂度



池化计算过程

图像分类神经网络

- 神经网络构建



利用AI平台加速训练实例

01
图像分类算法简介

02
图像分类算法云上加速

03
其他开源资源共享

图像分类（训练过程）

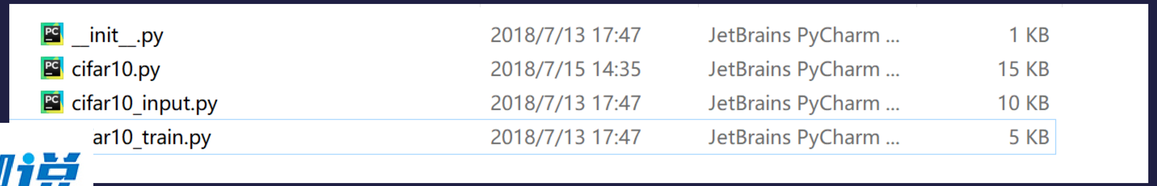
1 操作环境准备

- Linux环境或类Linux环境
- 安装docker
- 安装UFIle SDK
- 安装Ucloud AI SDK

2 代码和数据准备

- 准备训练代码
- 代码下载地址：

https://github.com/ucloud/uai-sdk/tree/master/examples/tensorflow/train/cifar_simple



图像分类（训练过程）

打包方法

- UAI-SDK提供的打包工具：tf_tool.py
- UAI-SDK自定义打包工具：base_tool.py
- 通过Dockerfile来打包镜像

以Dockerfile打包为例

1 创建Dockerfile

```
FROM uhub.service.ucloud.cn/uaishare/gpu_uaitrain_ubuntu-14.04_python-2.7.6_tensorflow-1.4.0:v1.0
ADD ./code/ /data/code/
```

- 设置基础镜像：uhub.service.ucloud.cn/uaishare/gpu_uaitrain_ubuntu-14.04_python-2.7.6_tensorflow-1.4.0:v1.0
- 将./code/拷贝到docker镜像的/data/code/目录下

2 编译镜像

```
r build –t uhub.service.ucloud.cn/uai_demo/cifar-train-gpu:v1.0 –f cifar.Dockerfile .
```

图像分类（训练过程）

4 上传镜像

- 将镜像上传至UHub容器镜像库中以备调用

```
Sudo docker push uhub.service.ucloud.cn/uai_demo/cifar-train-gpu:v1.0
```

5 下载并上传数据

- 下载数据

```
sudo docker run -it -v /data/data:/data/data  uhub.service.ucloud.cn/YOUR_UHUB_REGISTRY/cifar-train-cpu:v1.0
/bin/bash -c "python /data/data/download.py"
```

- 将数据上传至UFile平台

```
./filemgr-linux64 --action mput --bucket uai-demo --dir /data/data/  --prefix /cifar/train/
```

图像分类（训练过程）

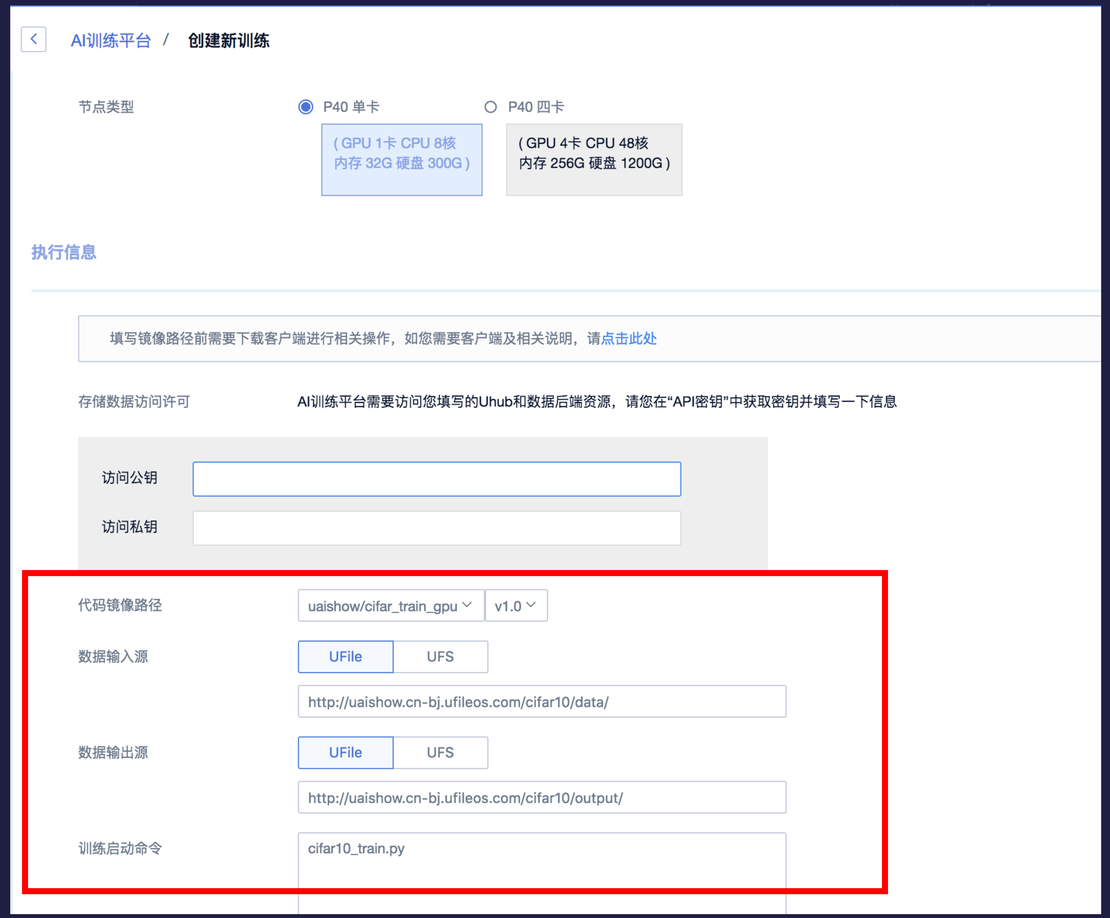
6 平台训练

- 选择相应的训练镜像：

  uhub.service.ucloud.cn/uai_demo/cifar-train-gpu:v1.0

- 设置输入数据路径
- 设置输出数据路径
- 训练命令

  /data/code/cifar10_train.py

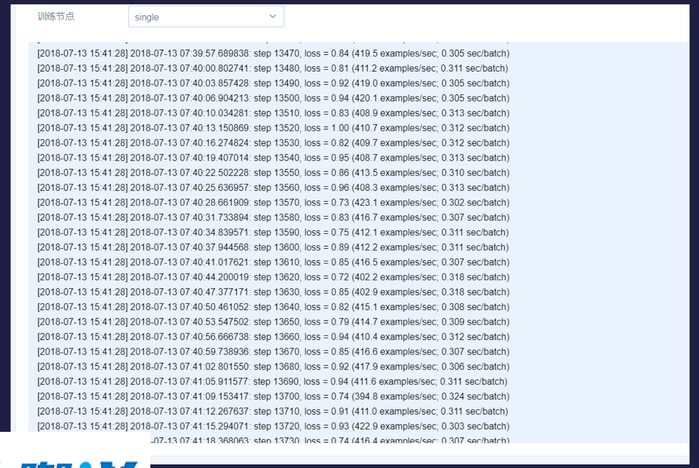


图像分类（训练过程）

7 获取训练结果

- 训练结果将自动上传到Ufile指定的输出中
- 通过命令行或UFile界面下载训练结果 例：

```
./filemgr-linux64 --action download --bucket uai-demo --key cifar_simple/train/output/model.ckpt-11428.data-00000-of-00002 --file /data/model.ckpt-11428.data-00000-of-00002
```



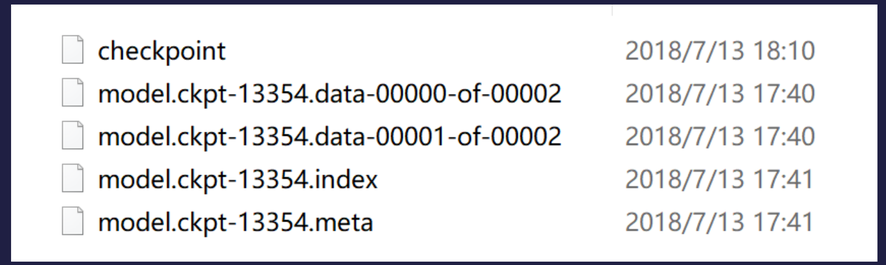
图像分类（在线推理）

1 操作环境准备

- Linux环境或类Linux环境
- 安装docker
- 安装UFIle SDK
- 安装Ucloud AI SDK

2 模型和推理服务代码准备

- 准备推理服务代码
- 准备模型文件



图像分类（在线推理）

推理服务代码:cifar_inference.py

1

```
#给出类别
label_dict={0:'airplane’,1:'automobile’,
2:'bird’,3:'cat',4:'deer',5:'dog',6:'frog’,
7:'horse’,8:'ship',9:'truck'}
#实现一个在线服务的类,
#继承了TFAiUcloudModel（TensorFlow 在线服务基类）
from uai.arch.tf_model import TFAiUcloudModel
    class cifarModel(TFAiUcloudModel):
```

2

```
#加载模型
def load_model(self):
        #创建graph
    sess = tf.Session()
    x = tf.placeholder(dtype=tf.float32, shape=[1, 24,
        24, 3], name='input’)
    pred = tf.argmax(cifar10.inference(x),axis=1)
    #读入模型文件， model_dir路径在初始化时从conf.json中获取
    saver = tf.train.Saver()
    params_file = tf.train.latest_checkpoint(self.model_dir)
    saver.restore(sess, params_file)
    #将执行推理所需的sess、x、y_三个变量保存到
    #cifarModel.output全局变量中
    self.output['sess'] = sess
    self.output['x'] = x
    self.output['y_'] = pred
```

图像分类（在线推理）

推理服务代码:cifar_inference.py

3

```
def execute(self, data, batch_size):
    #从cifarModel.output全局变量中获取sess、x、y_三个变量
    sess = self.output['sess']
    x = self.output['x']
    y_ = self.output['y_']
    #从data获取batching的请求数据
    ret = []
```

4

```
for i in range(batch_size):
    image = Image.open(data[i])
    image = cv2.cvtColor(np.asarray(image),cv2.COLOR_RGB2BGR)
    image = cv2.resize(image, (24, 24))
    mean=np.mean(image)
    std=np.std(image)
    image=(image-mean)/max(std,1/np.sqrt(image.size))
    image = np.expand_dims(image, axis=0).astype(np.float32)
    #请求推理操作：
    preds = sess.run(y_, feed_dict={x: image})
    pred_label=label_dict[preds[0]]
    ret.append(pred_label)
return ret
```

图像分类（在线推理）

打包方法

- UAI-SDK提供的打包工具：uai_tool.py
- 通过Dockerfile来打包镜像

以Dockerfile打包为例

```
FROM uhub.service.ucloud.cn/uaishare/cpu_uaiservice_ubuntu-14.04_python-2.7.6_tensorflow-1.4.0:v1.2
EXPOSE 8080
ADD ./inference/ /ai-ucloud-client-django/
ADD ./code/ /ai-ucloud-client-django/
ADD ./cifar.conf  /ai-ucloud-client-django/conf.json
ENV UAI_SERVICE_CONFIG /ai-ucloud-client-django/conf.json
CMD cd /ai-ucloud-client-django && gunicorn -c gunicorn.conf.py httpserver.wsgi
```

图像分类（在线推理）

本地测试

- 在本地docker中运行：sudo docker run -it -p 8080:8080 uhub.service.ucloud.cn/uai_demo/cifar_infer_simple:v1.0
- 使用图片cat.jpg测试：curl -X POST http://localhost:8080/service -T cat.jpg

平台测试

- 在UAI-Inference中创建新任务，选择我们push上去的镜像进行部署，启动；
- 使用图片cat.jpg测试：curl -X POST http://<url>/service -T cat.jpg（url是我们在平台上得到的地址）

利用AI平台加速训练实例

01 图像分类算法简介

02 图像分类算法云上加速

03 其他开源资源共享

开源镜像仓库

推理案例镜像仓库

https://docs.ucloud.cn/ai/uai-inference/general/examples

UAI Inference 开源案例一览

TensorFlow

- mnist: https://github.com/ucloud/uai-sdk/tree/master/examples/tensorflow/inference/mnist_1.1
- east(CPU,GPU): https://github.com/ucloud/uai-sdk/tree/master/examples/tensorflow/inference/east
- facenet: https://github.com/ucloud/uai-sdk/tree/master/examples/tensorflow/inference/facenet
- text-classification: https://github.com/ucloud/uai-sdk/tree/master/examples/tensorflow/inference/text-classification-ch
- inception(tf-serve compatable): https://github.com/ucloud/uai-sdk/tree/master/examples/tensorflow/inference/tf-serving/inception
- wide&deep(tf-serve compatable): https://github.com/ucloud/uai-sdk/tree/master/examples/tensorflow/inference/tf-serving/wide_deep

MXNet

- mnist: https://github.com/ucloud/uai-sdk/tree/master/examples/mxnet/mnist

Caffe

- mnist: https://github.com/ucloud/uai-sdk/tree/master/examples/caffe/inference/mnist
- mtcnn: https://github.com/ucloud/uai-sdk/tree/master/examples/caffe/inference/mtcnn
- ctpn(Intel Caffe): https://github.com/ucloud/uai-sdk/tree/master/examples/caffe/inference/intel-caffe/ctpn
- rfcn(Intel Caffe): https://github.com/ucloud/uai-sdk/tree/master/examples/caffe/inference/intel-

ub.com/ucloud/uai-sdk/tree/master/examples/caffe/inference/rfcn

训练案例镜像仓库

https://docs.ucloud.cn/ai/uai-train/general/examples

UAI Train 开源案例一览

TensorFlow

- mnist_summary: https://github.com/ucloud/uai-sdk/tree/master/examples/tensorflow/train/mnist_summary_1.1
- cifar: https://github.com/ucloud/uai-sdk/tree/master/examples/tensorflow/train/cifar
- imagenet(resnet): https://github.com/ucloud/uai-sdk/tree/master/examples/tensorflow/train/imagenet
- slim: https://github.com/ucloud/uai-sdk/tree/master/examples/tensorflow/train/slim
- retrain: https://github.com/ucloud/uai-sdk/tree/master/examples/tensorflow/train/retrain
- east: https://github.com/ucloud/uai-sdk/tree/master/examples/tensorflow/train/east
- object-detection: https://github.com/ucloud/uai-sdk/tree/master/examples/tensorflow/train/object-detection
- im2txt: https://github.com/ucloud/uai-sdk/tree/master/examples/tensorflow/train/im2txt
- test-classification-ch: https://github.com/ucloud/uai-sdk/tree/master/examples/tensorflow/train/text-classification-ch
- wide&deep: https://github.com/ucloud/uai-sdk/tree/master/examples/tensorflow/train/wide-deep

MXNet

- mnist: https://github.com/ucloud/uai-sdk/tree/master/examples/mxnet/train/mnist
- cifar: https://github.com/ucloud/uai-sdk/tree/master/examples/mxnet/train/cifar
- imagenet(resnet): https://github.com/ucloud/uai-sdk/tree/master/examples/mxnet/train/imagenet

Caffe

- mnist: https://github.com/ucloud/uai-sdk/tree/master/examples/caffe/train/mnist
- imagenet(resnet): https://github.com/ucloud/uai-sdk/tree/master/examples/caffe/train/imagenet/code
- faster-rcnn:https://github.com/ucloud/uai-sdk/tree/master/examples/caffe/train/faster-rcnn
- rfcn:https://github.com/ucloud/uai-sdk/tree/master/examples/caffe/train/rfcn

Pytorch

- mnist: https://github.com/ucloud/uai-sdk/tree/master/examples/pytorch/train/mnist

开源github及SDK

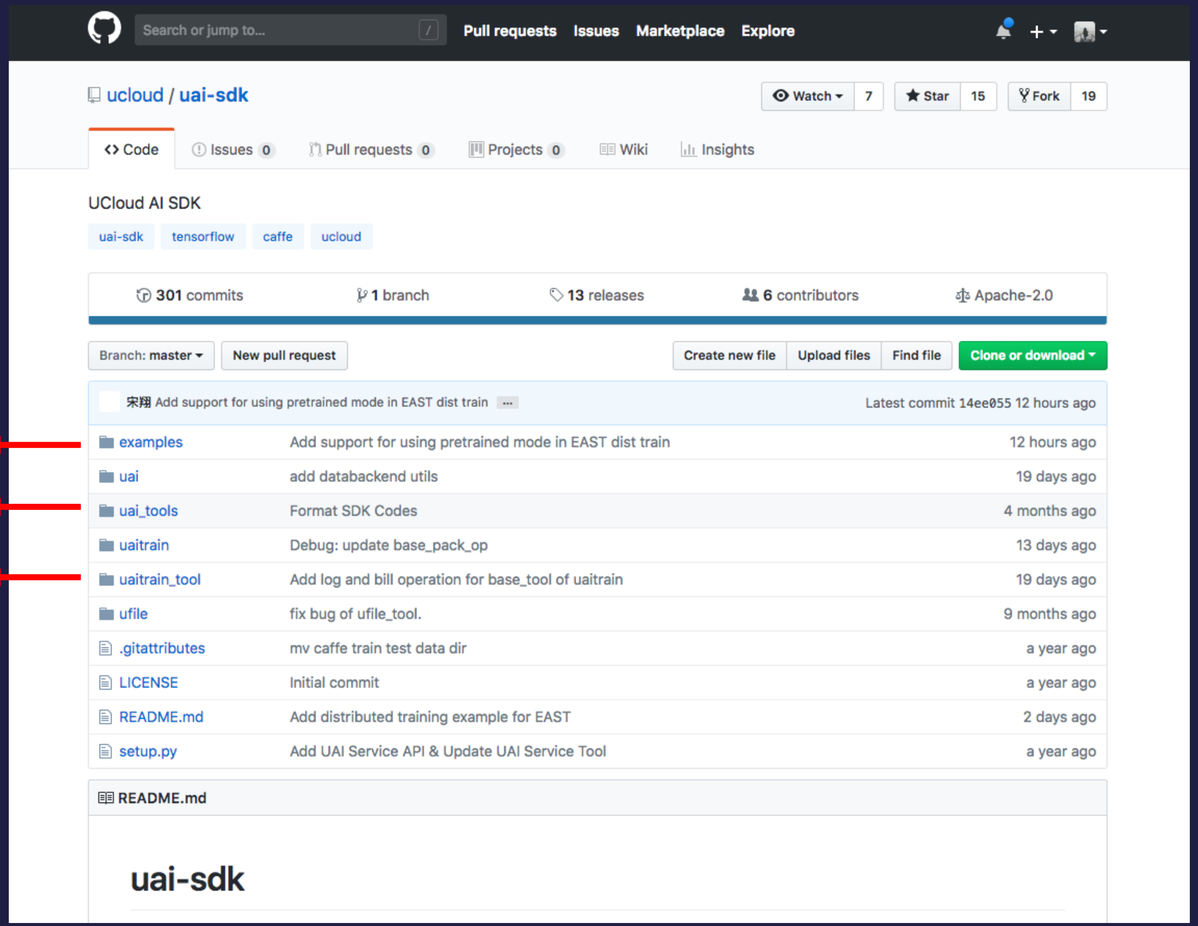
github开源项目

https://github.com/ucloud/uai-sdk

开源案例源码（example）

推理打包工具（uai_tools）

训练打包工具（uaitrain_tools）



悟有所值
Ucan 下午茶
Ucan Afternoon Tea

THANKS

UCLOUD 专业云计算服务商

IT大咖说 知识共享平台