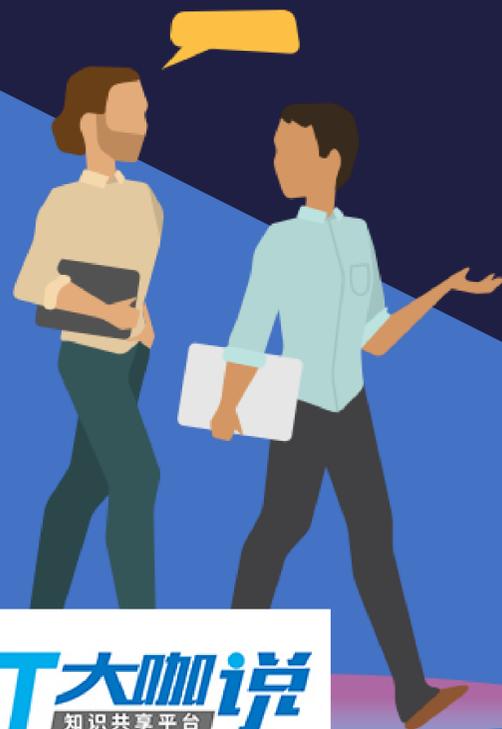


AI 公有云实践

范融 UCloud深度学习工程师



目录

01

为什么要建设AI
公有云平台

02

UCloud AI基础平
台架构

03

利用AI平台加速训
练实例

01

为什么要建设AI公有云平台

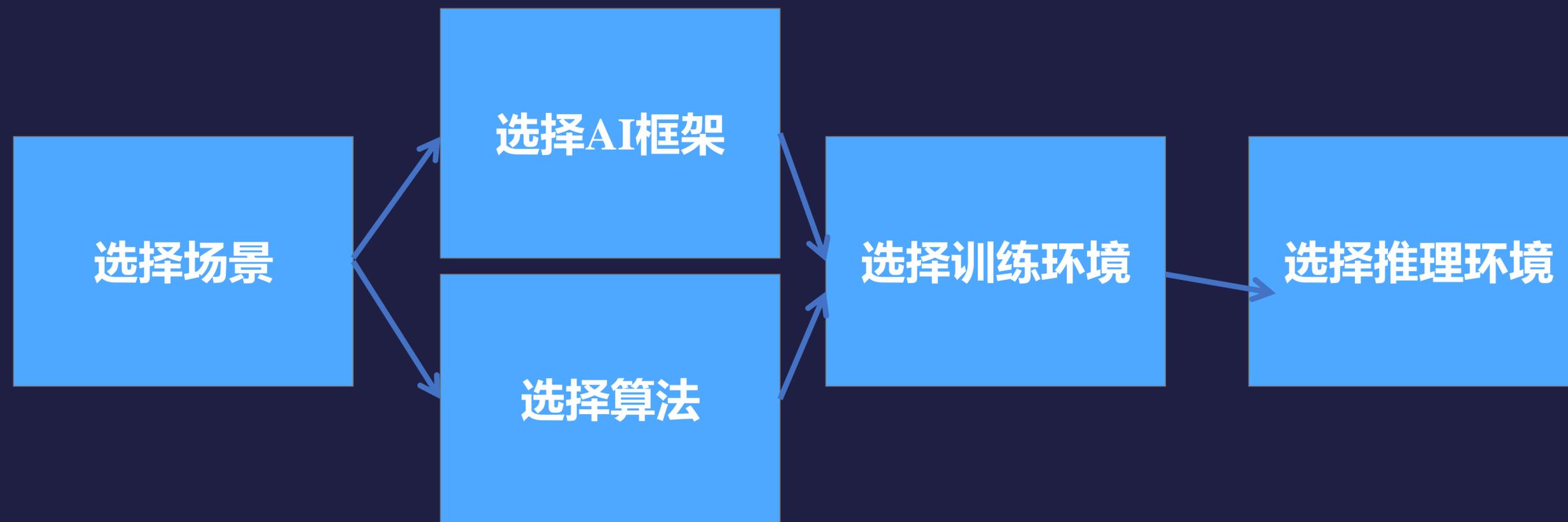
AI运用的各类场景



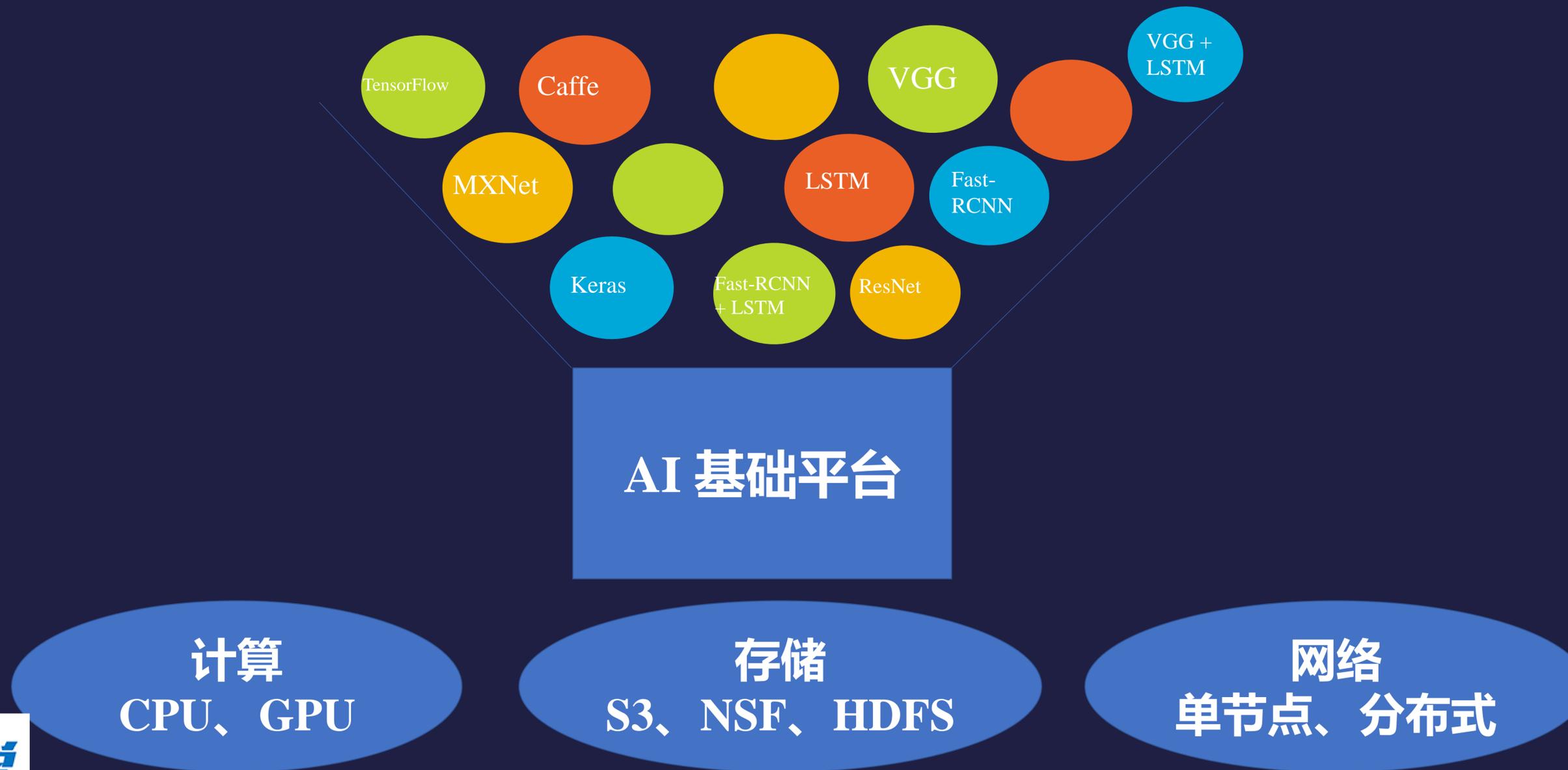
AI技术的各类选择



AI项目研发周期

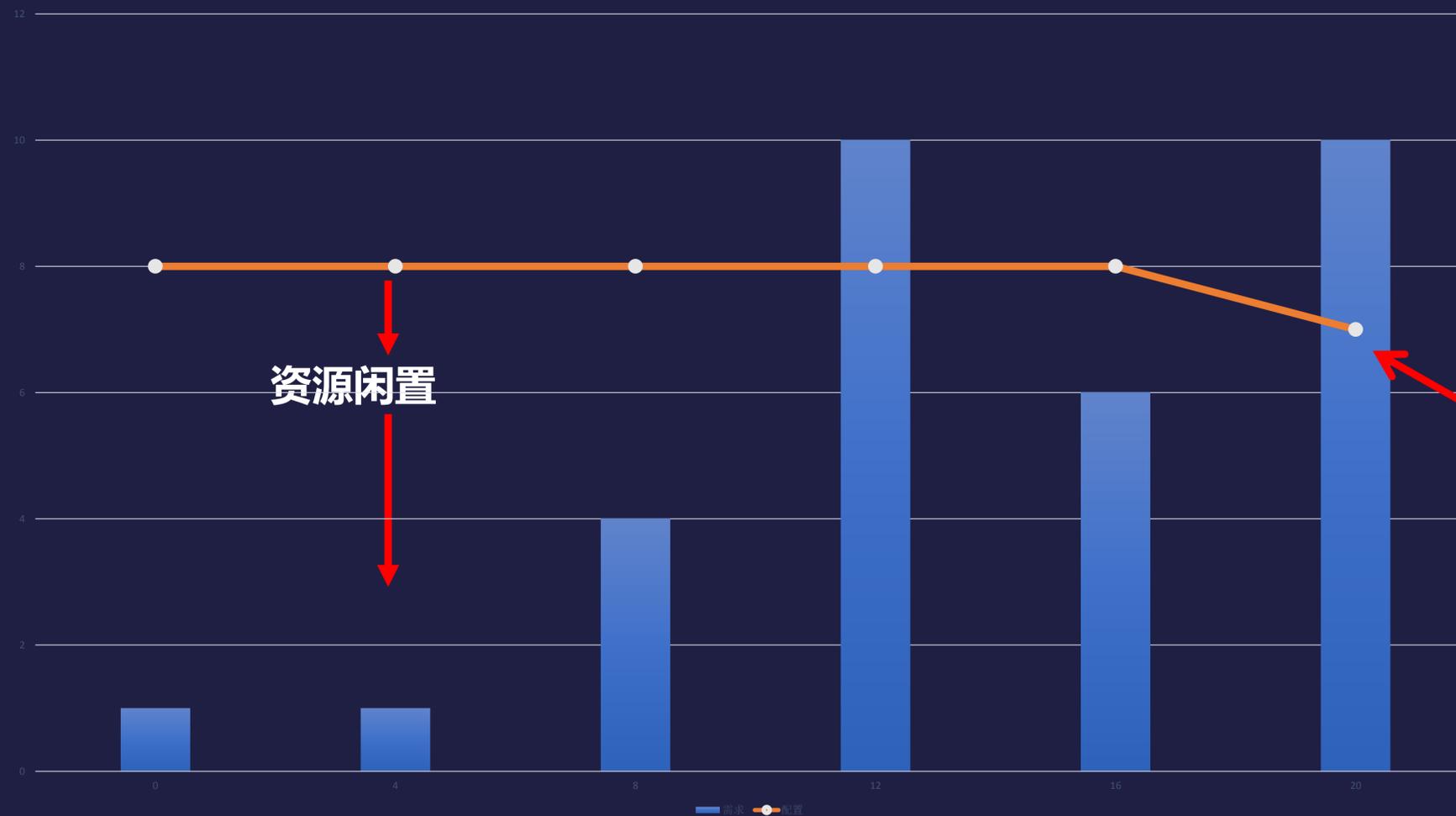


AI平台需求（技术平台）



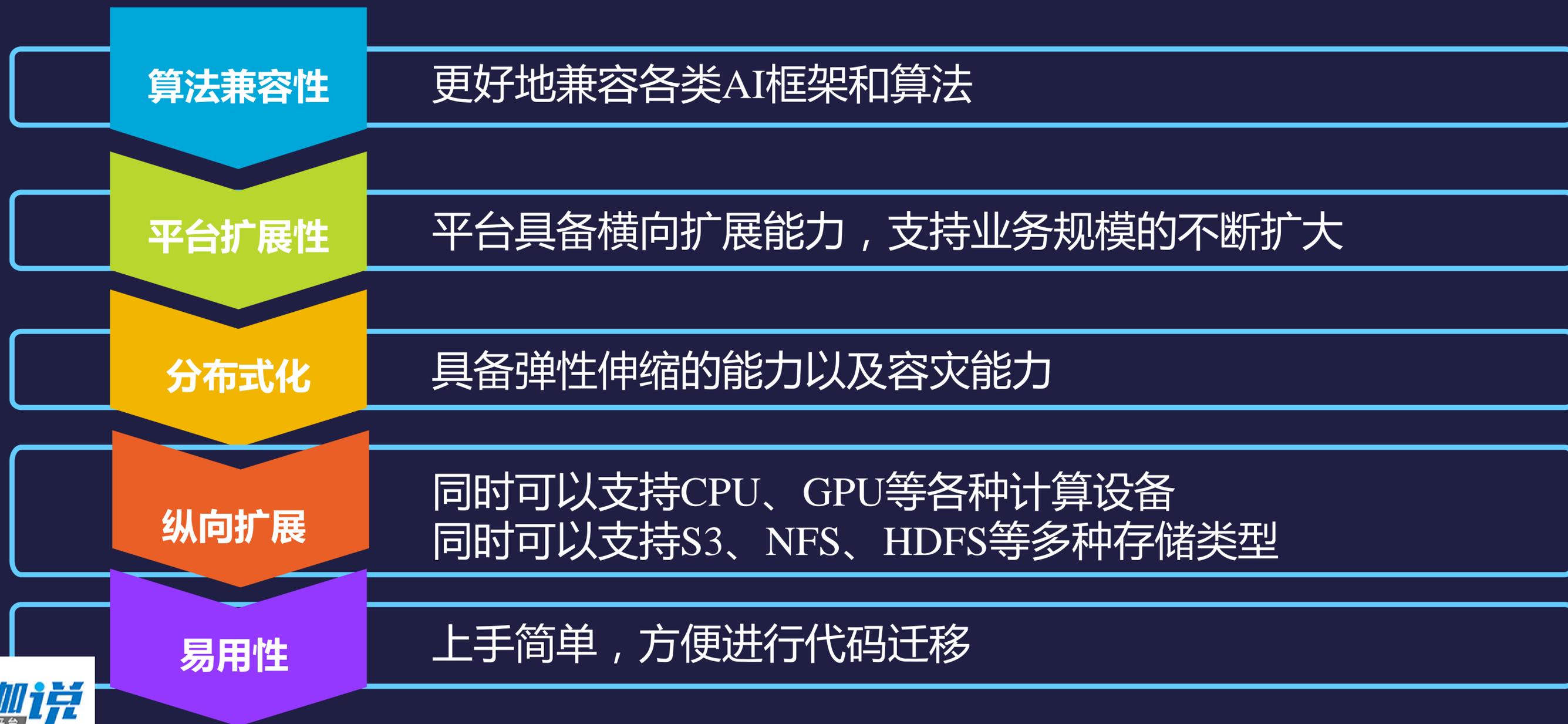
AI平台需求（项目需求）

项目组资源需求曲线



计算资源
损坏, 资源
不足++

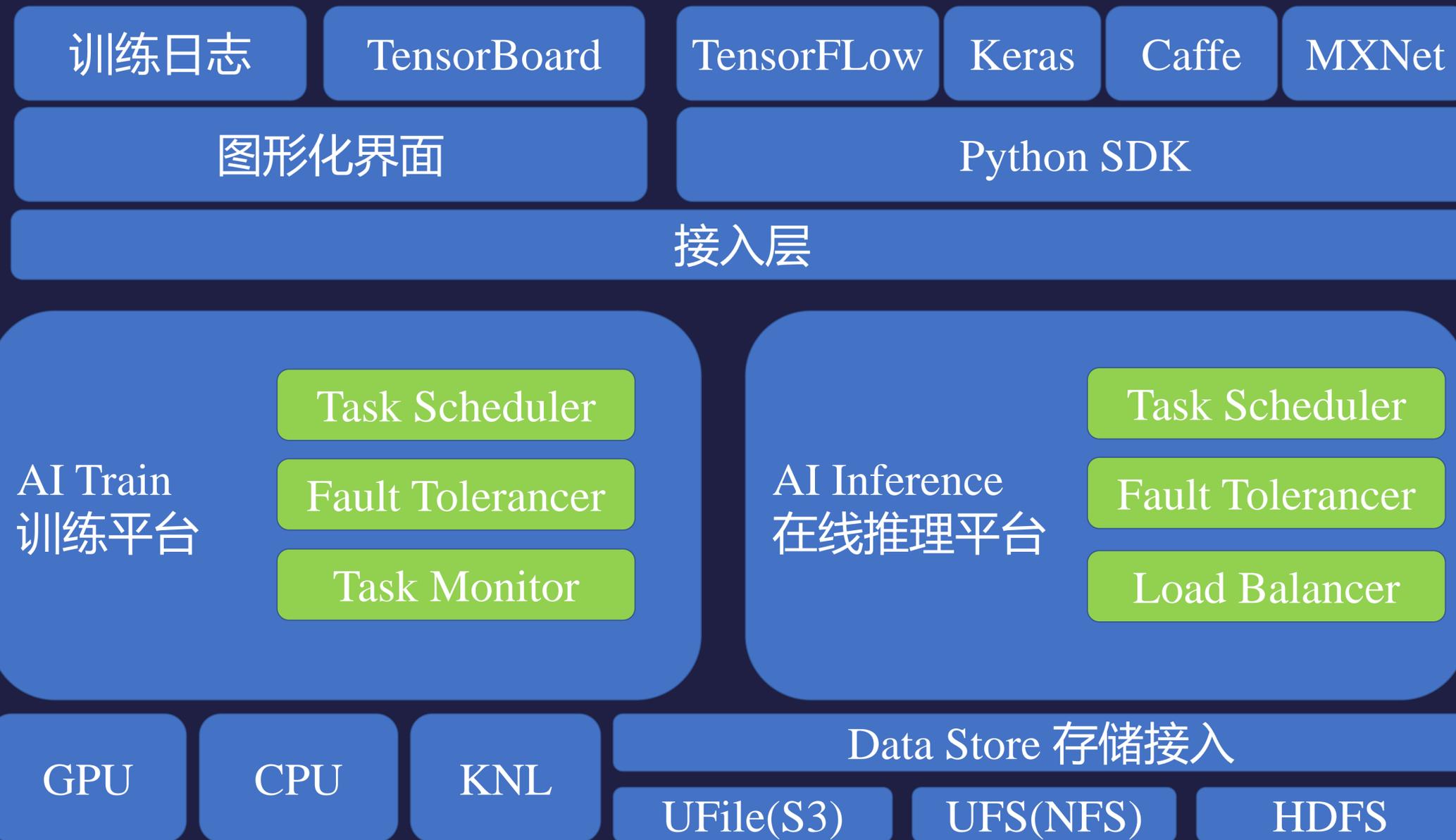
AI平台的兼容性



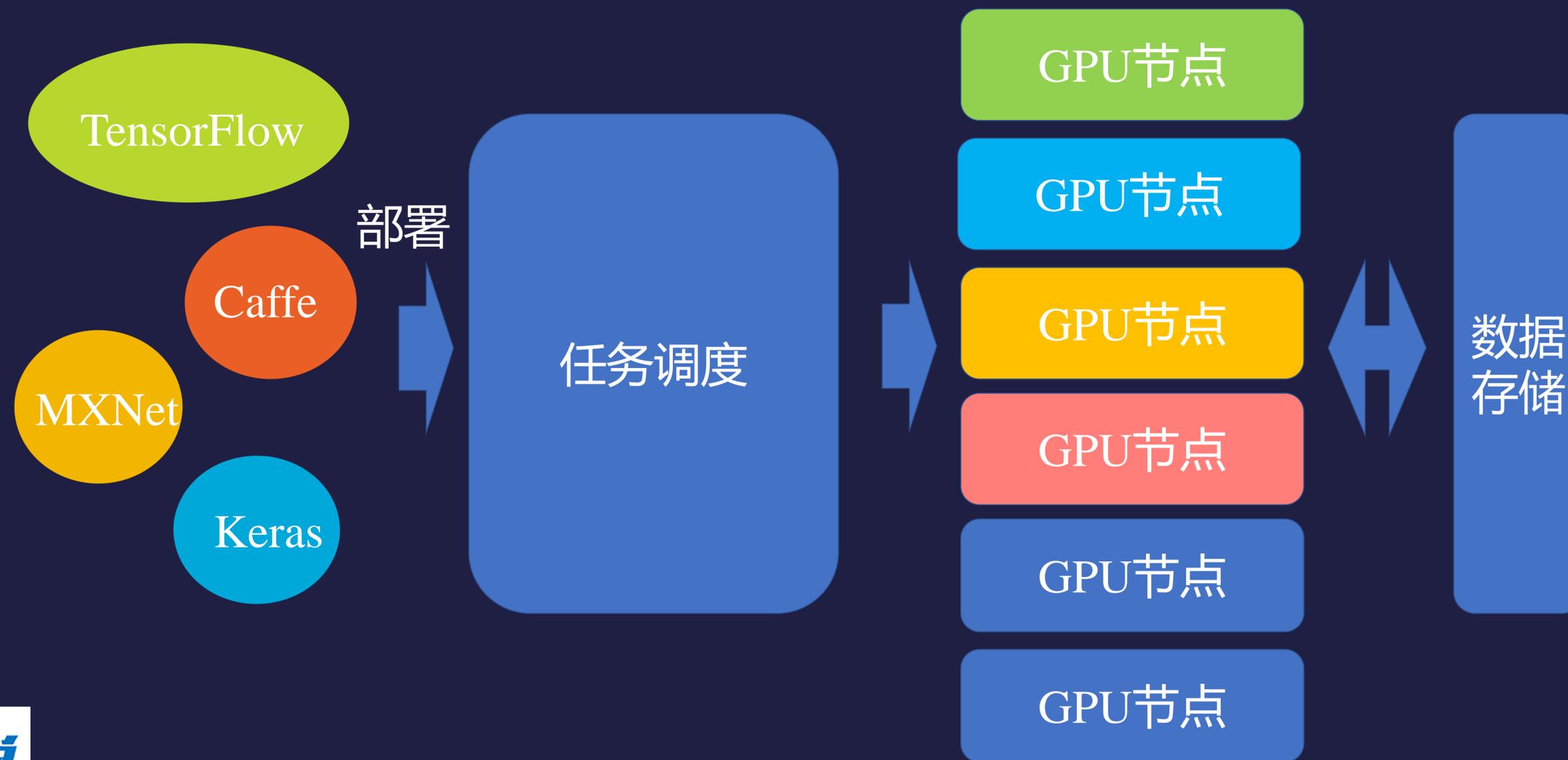
02

UCloud AI基础平台架构

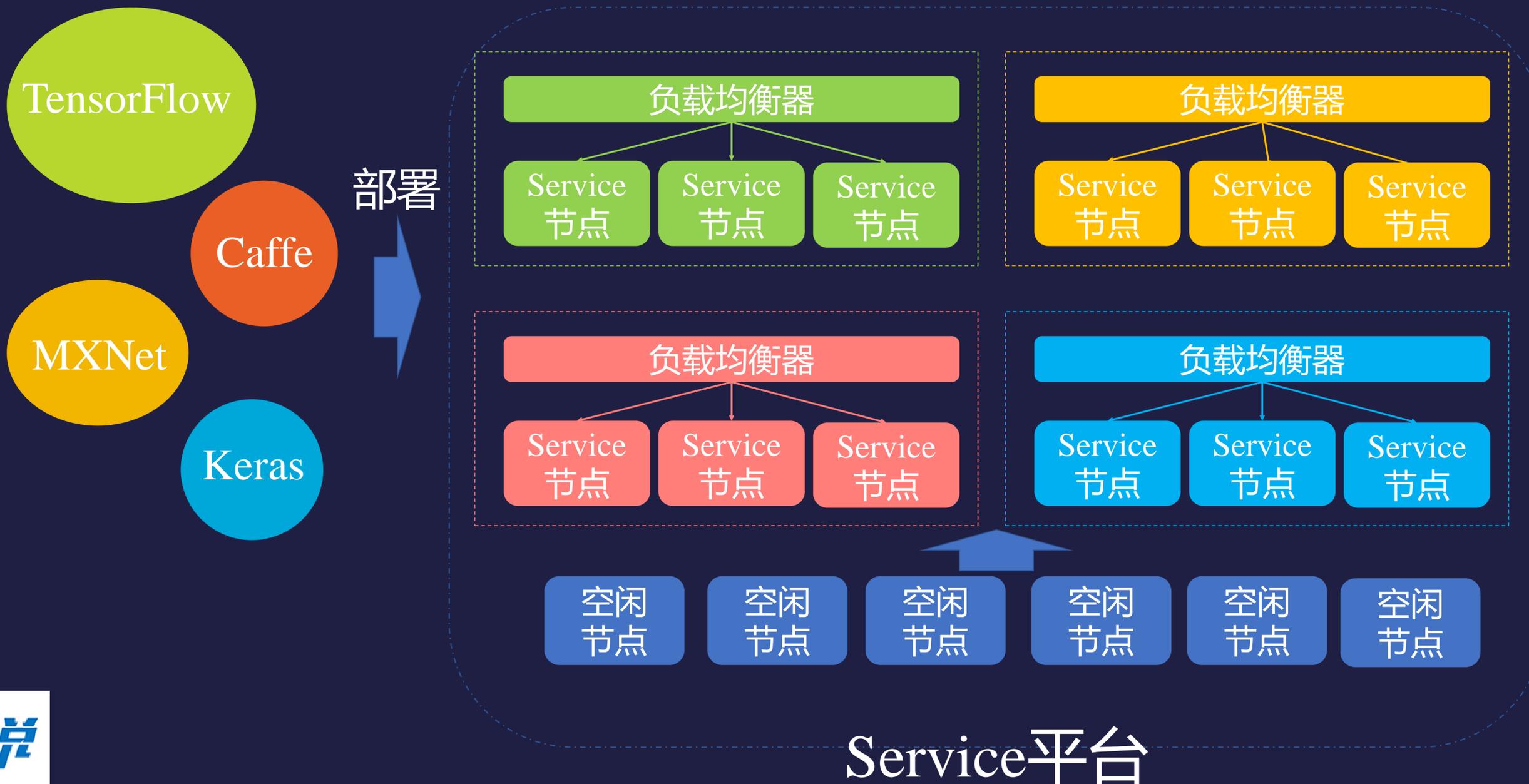
UCloud AI平台整体架构



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）

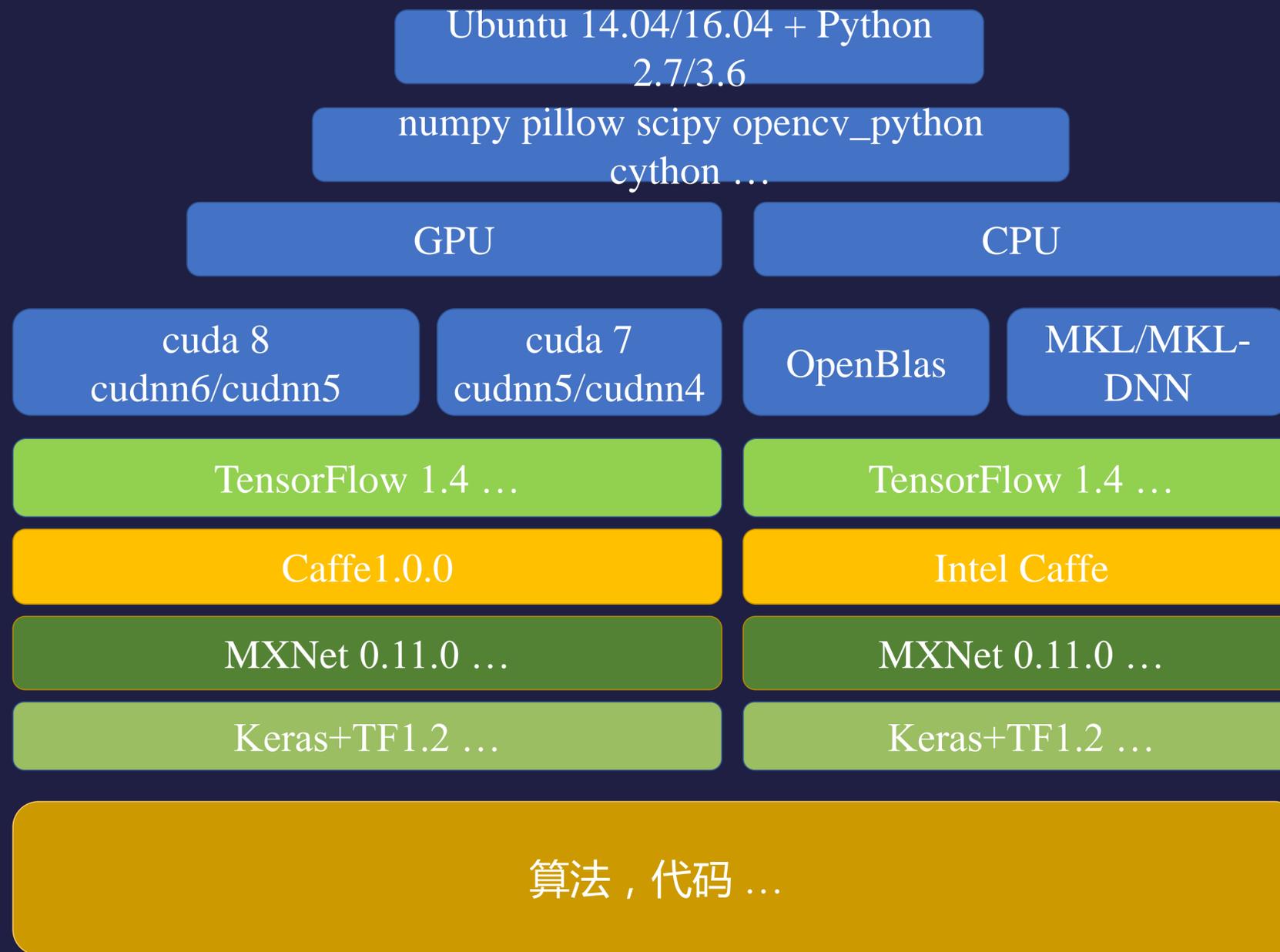
- 无需用户考虑
- 自动支持GPU加速

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



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

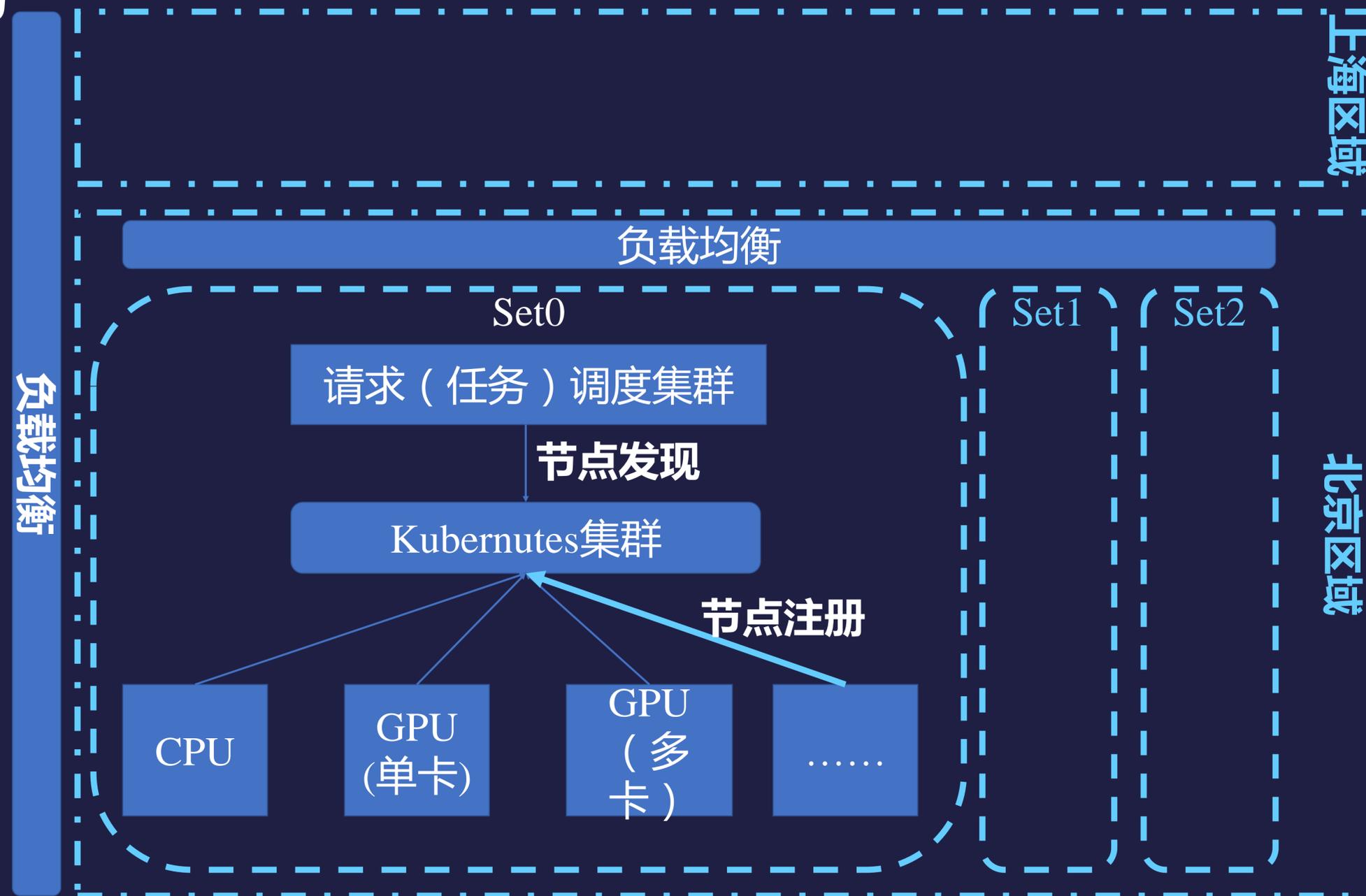
弹性

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



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

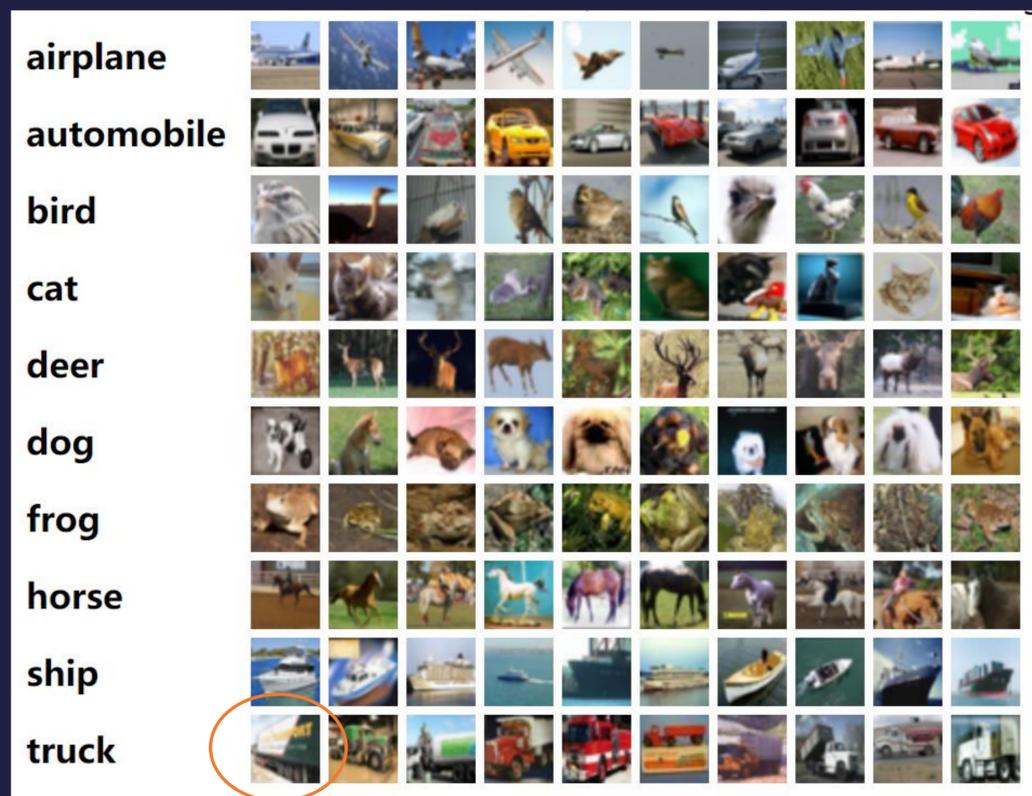
02

图像分类算法
云上加速

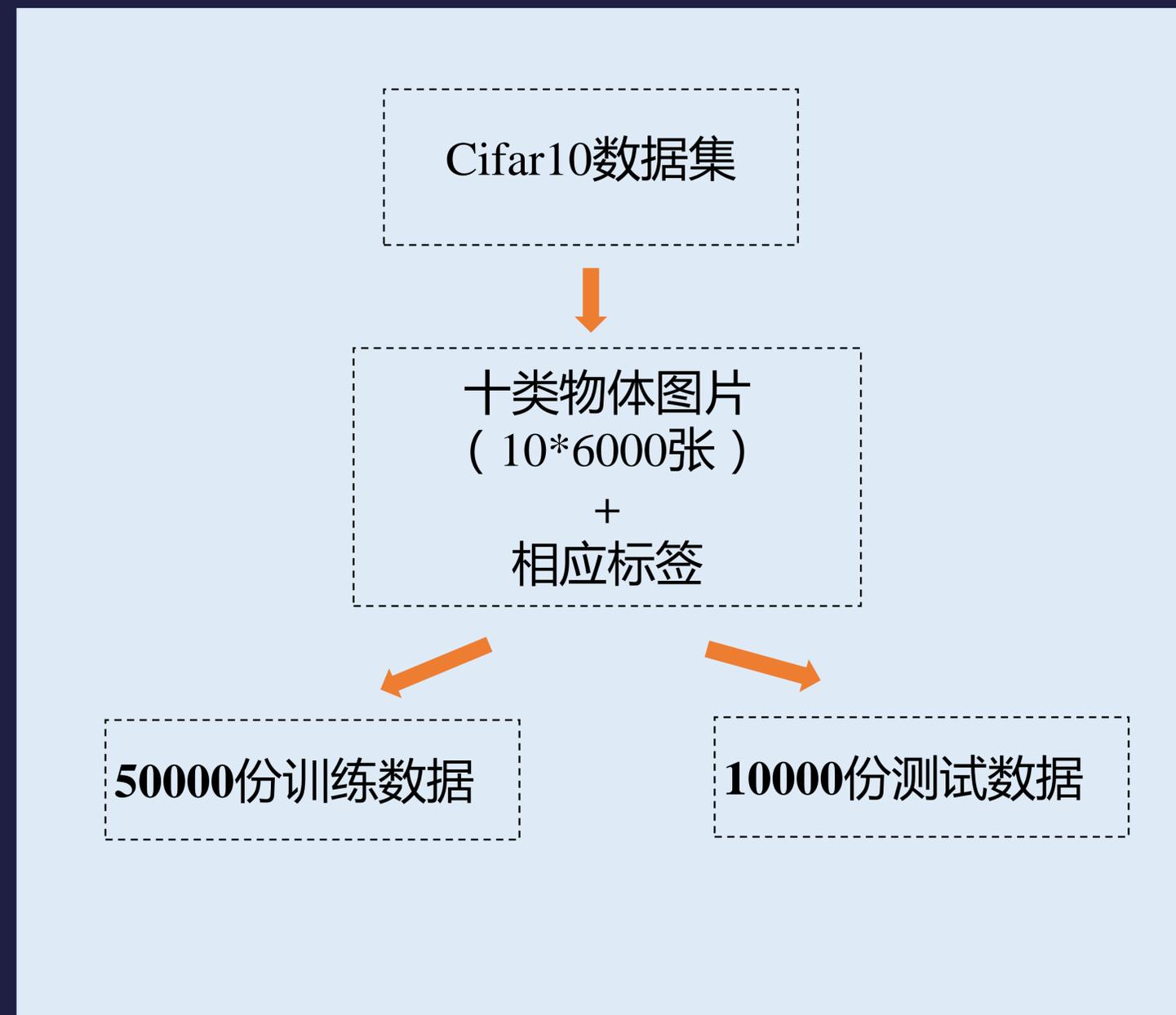
03

其他开源资源共享

图像分类案例——cifar数据集

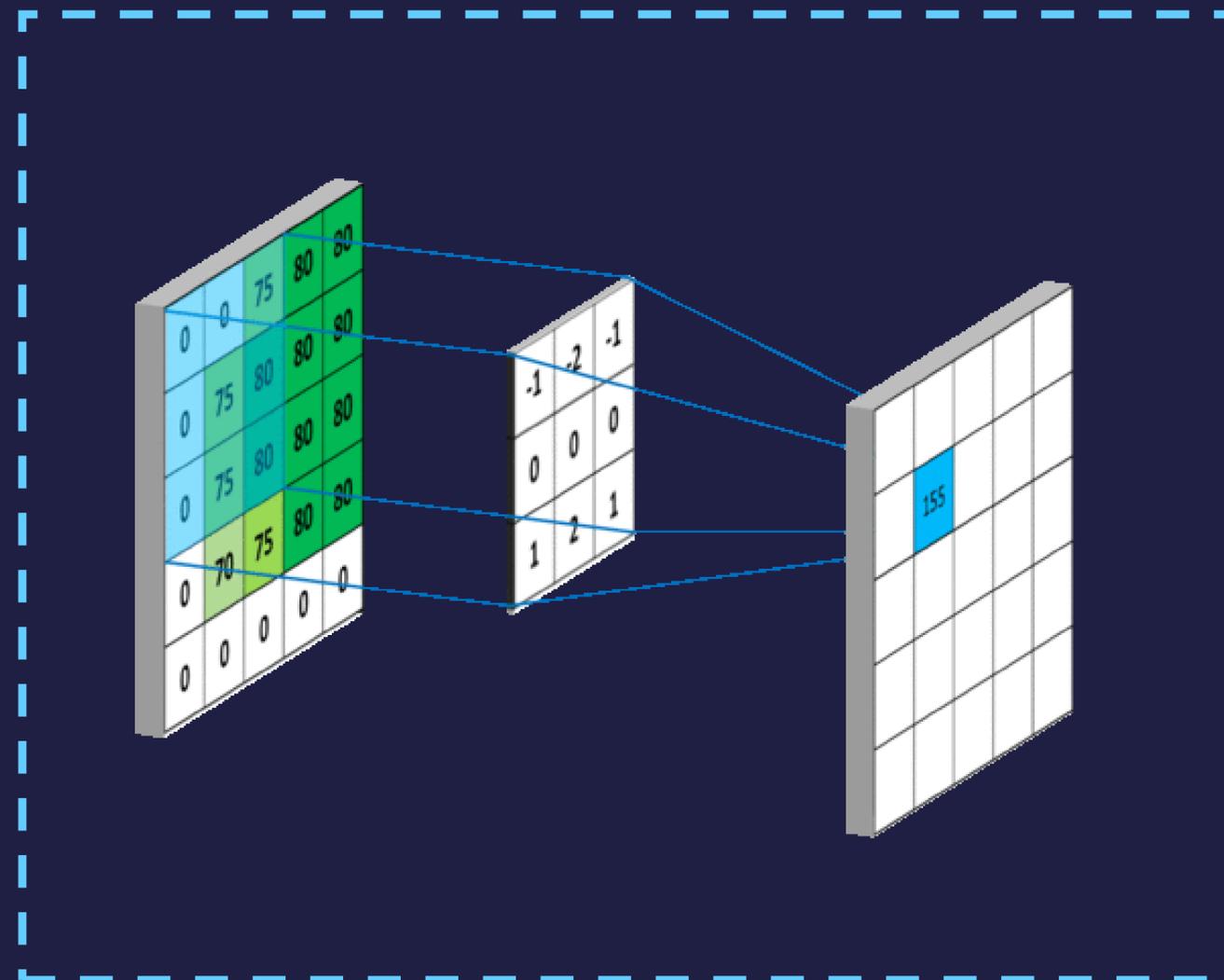


大小为32*32
的RGB图像



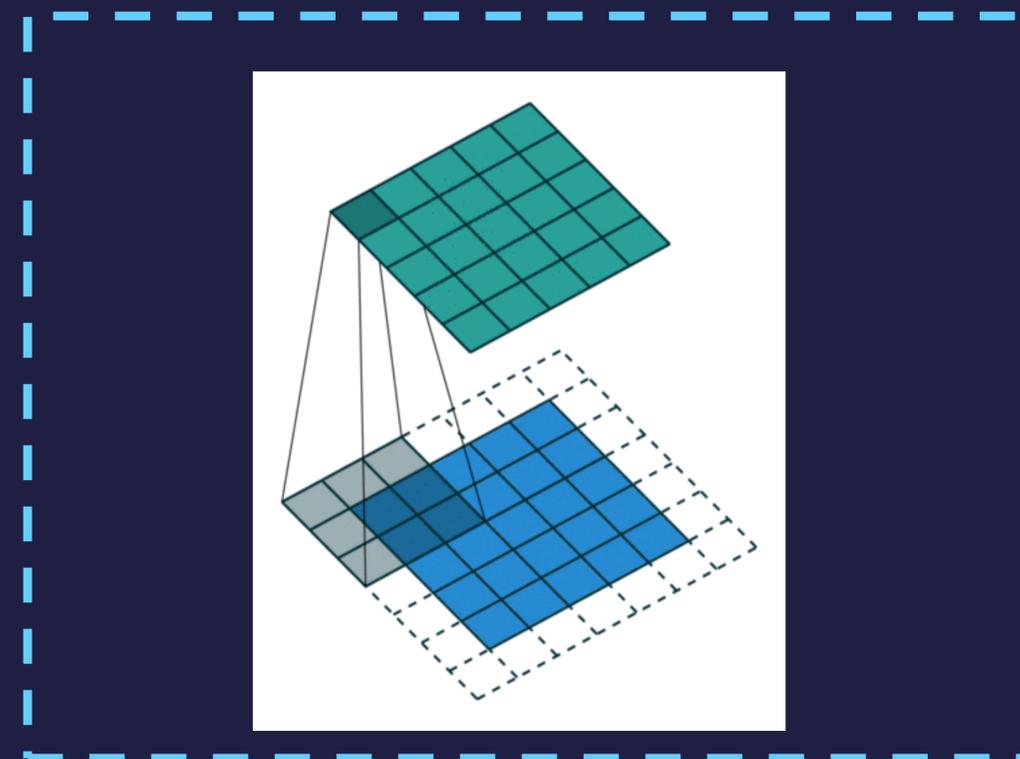
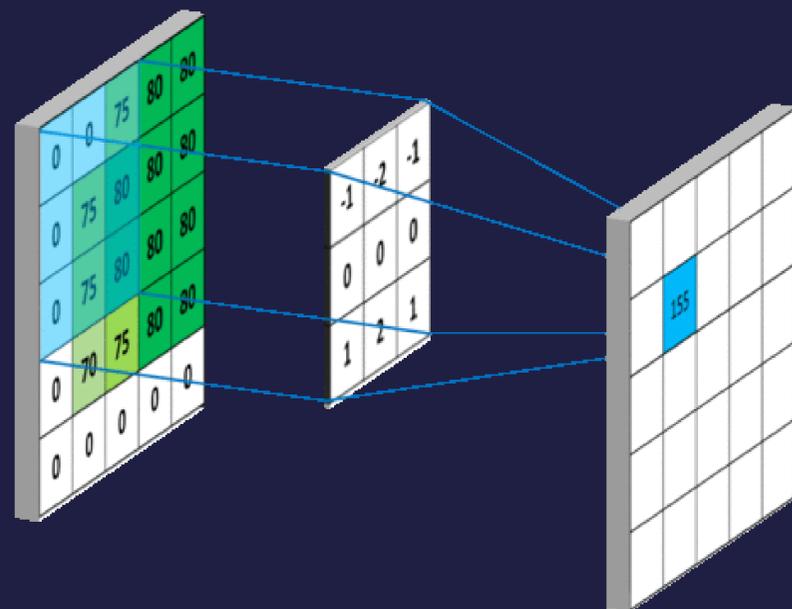
图像分类神经网络

- 卷积计算



图像分类神经网络

• 卷积计算



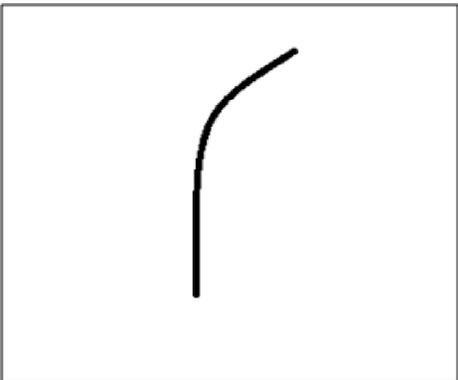
图像分类神经网络

• 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

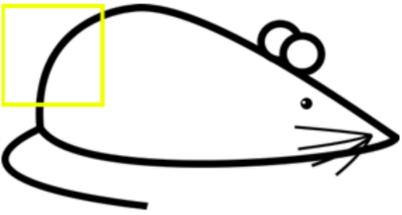


Visualization of a curve detector filter

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

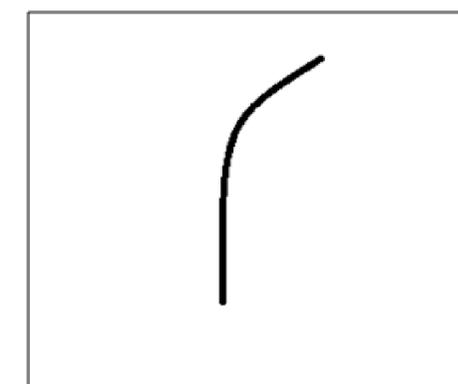
• 卷积计算

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。

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

卷积核



Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

Pixel representation of 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

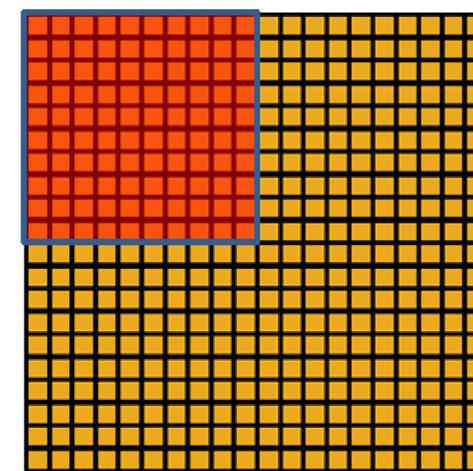
Multiplication and Summation = 0

卷积计算

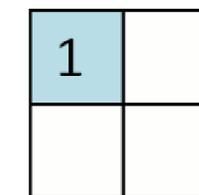
图像分类神经网络

• 池化计算

池化运算
提取主要特征；
减小计算复杂度



Convolved
feature

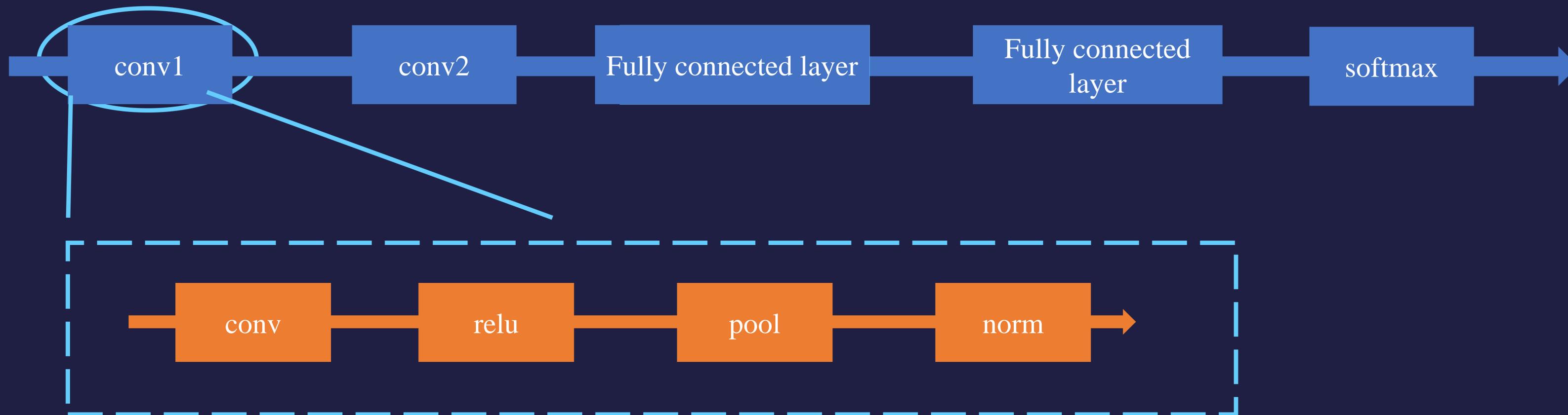


Pooled
feature

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

 __init__.py	2018/7/13 17:47	JetBrains PyCharm ...	1 KB
 cifar10.py	2018/7/15 14:35	JetBrains PyCharm ...	15 KB
 cifar10_input.py	2018/7/13 17:47	JetBrains PyCharm ...	10 KB
ar10_train.py	2018/7/13 17:47	JetBrains PyCharm ...	5 KB

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

```
docker 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

AI训练平台 / 创建新训练

节点类型 P40 单卡 P40 四卡

(GPU 1卡 CPU 8核 内存 32G 硬盘 300G) (GPU 4卡 CPU 48核 内存 256G 硬盘 1200G)

执行信息

填写镜像路径前需要下载客户端进行相关操作，如您需要客户端及相关说明，请点击[此处](#)

存储数据访问许可 AI训练平台需要访问您填写的Uhub和数据后端资源，请您在“API密钥”中获取密钥并填写一下信息

访问公钥

访问私钥

代码镜像路径 uaishow/cifar_train_gpu v1.0

数据输入源 UFile UFS

数据输出源 UFile UFS

训练启动命令

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
```

```
训练节点 single
[2018-07-13 15:41:28] 2018-07-13 07:39:57.689838: step 13470, loss = 0.84 (419.5 examples/sec, 0.305 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:00.802741: step 13480, loss = 0.81 (411.2 examples/sec, 0.311 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:03.857428: step 13490, loss = 0.92 (419.0 examples/sec, 0.305 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:06.904213: step 13500, loss = 0.94 (420.1 examples/sec, 0.305 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:10.034281: step 13510, loss = 0.83 (408.9 examples/sec, 0.313 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:13.150869: step 13520, loss = 1.00 (410.7 examples/sec, 0.312 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:16.274824: step 13530, loss = 0.82 (409.7 examples/sec, 0.312 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:19.407014: step 13540, loss = 0.95 (408.7 examples/sec, 0.313 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:22.502228: step 13550, loss = 0.86 (413.5 examples/sec, 0.310 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:25.636957: step 13560, loss = 0.96 (408.3 examples/sec, 0.313 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:28.661909: step 13570, loss = 0.73 (423.1 examples/sec, 0.302 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:31.733894: step 13580, loss = 0.83 (416.7 examples/sec, 0.307 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:34.839571: step 13590, loss = 0.75 (412.1 examples/sec, 0.311 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:37.944568: step 13600, loss = 0.89 (412.2 examples/sec, 0.311 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:41.017621: step 13610, loss = 0.85 (416.5 examples/sec, 0.307 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:44.200019: step 13620, loss = 0.72 (402.2 examples/sec, 0.318 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:47.377171: step 13630, loss = 0.85 (402.9 examples/sec, 0.318 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:50.461052: step 13640, loss = 0.82 (415.1 examples/sec, 0.308 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:53.547502: step 13650, loss = 0.79 (414.7 examples/sec, 0.309 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:56.666738: step 13660, loss = 0.94 (410.4 examples/sec, 0.312 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:40:59.738936: step 13670, loss = 0.85 (416.6 examples/sec, 0.307 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:41:02.801550: step 13680, loss = 0.92 (417.9 examples/sec, 0.306 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:41:05.911577: step 13690, loss = 0.94 (411.6 examples/sec, 0.311 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:41:09.153417: step 13700, loss = 0.74 (394.8 examples/sec, 0.324 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:41:12.267637: step 13710, loss = 0.91 (411.0 examples/sec, 0.311 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:41:15.294071: step 13720, loss = 0.93 (422.9 examples/sec, 0.303 sec/batch)
[2018-07-13 15:41:28] 2018-07-13 07:41:18.368063: step 13730, loss = 0.74 (416.4 examples/sec, 0.307 sec/batch)
```

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

checkpoint	2018/7/13 18:10
model.ckpt-13354.data-00000-of-00002	2018/7/13 17:40
model.ckpt-13354.data-00001-of-00002	2018/7/13 17:40
model.ckpt-13354.index	2018/7/13 17:41
model.ckpt-13354.meta	2018/7/13 17:41

图像分类（在线推理）

推理服务代码:cifar_inference.py

#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):
```

1

#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
```

2

图像分类（在线推理）

推理服务代码:cifar_inference.py

```
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 = []
```

3

```
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
```

4

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/uaiservice/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](https://github.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

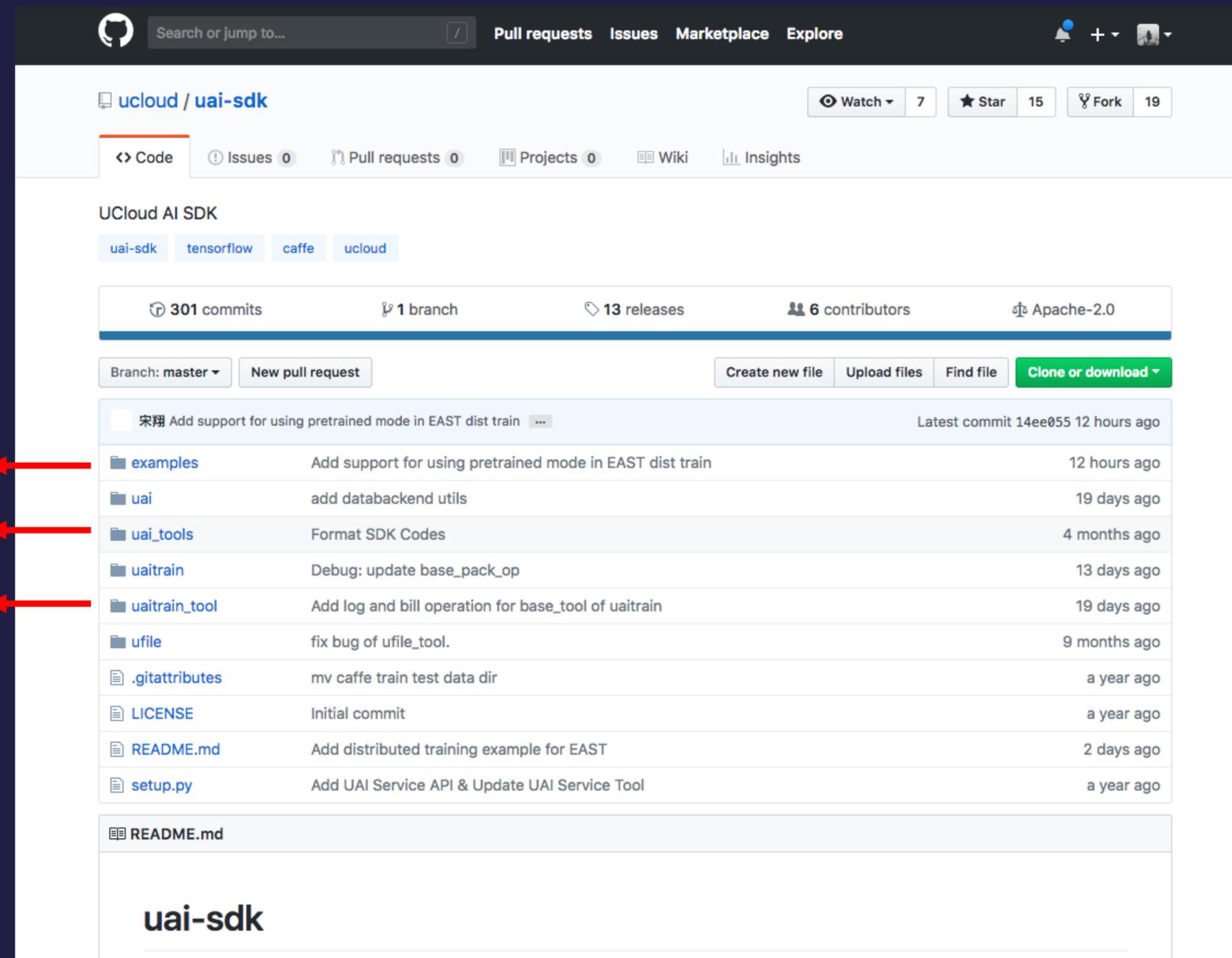
github开源项目

<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)



THANKS

