

## 奇虎360正式开源深度学习调度平台XLearning

本文系奇虎360系统部相关工程师投稿。

近两年人工智能技术发展迅速，以Google开源的TensorFlow为代表的各种深度学习框架层出不穷。为了方便算法工程师使用各类深度学习技术，减少繁杂的诸如运行环境部署运维等工作，提升GPU等硬件资源利用率，节省硬件投入成本，奇虎360系统部大数据团队与人工智能研究院联合开发了深度学习调度平台——XLearning。

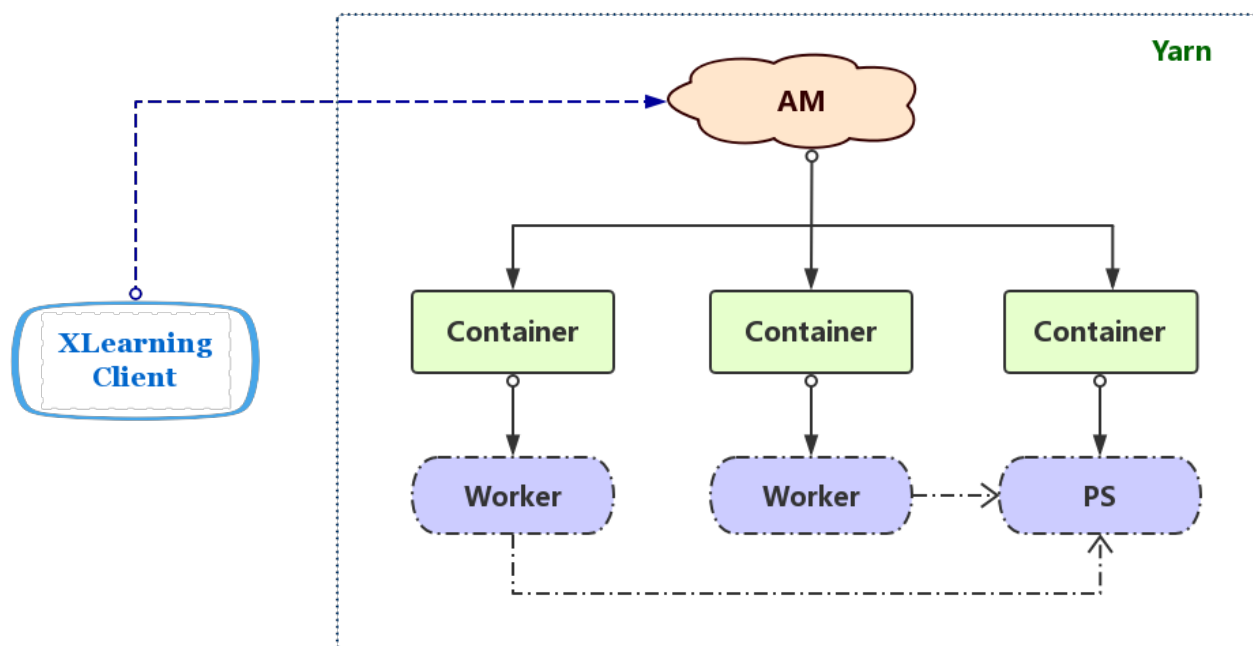


XLearning平台将大数据与深度学习相融合，基于Hadoop Yarn完成了对TensorFlow、MXNet、Caffe、Theano、PyTorch、Keras、XGBoost等常用深度学习框架的集成，是典型的“AI on Hadoop”的实现。XLearning从今年（2017）4月份正式开发上线运行，经多次版本迭代更新，为各学习框架的使用者提供了统一、稳定的调度平台，实现了资源共享，极大的提高了资源利用率，并且具有良好的扩展性和兼容性。目前已经在公司搜索、人工智能研究院、商业化、数据中心等业务部门广泛使用。

XLearning项目开源地址：

[GitHub地址](#)[Gitee地址](#)[g-xlearning-dev@360.cn](mailto:g-xlearning-dev@360.cn)加入QQ群:588356340

### XLearning架构



- Client : XLearning客户端，负责启动作业及获取作业执行状态;
- ApplicationMaster ( AM )  
:负责输入数据分片、启动及管理Container、执行日志保存等;
- Container : 作业的实际执行者，负责启动Worker或PS ( Parameter Server ) 进程，监控并向AM汇报进程状态，上传作业的输出等。对于TensorFlow类型作业，还负责启动TensorBoard服务。

## XLearning功能特性

XLearning虽然架构简洁，但具有丰富的功能方便用户进行模型训练，并依托于Yarn提供有作业资源的统一管理。

### 支持多种深度学习框架

XLearning支持TensorFlow、MXNet分布式和单机模式，支持所有的单机模式的深度学习框架，如Caffe、Theano、PyTorch等。对于同一个深度学习框架支持多版本和自定义版本，满足用户个性化需求，不受限于集群机器上各学习框架的安装版本。

### 基于HDFS的统一数据管理

XLearning提供多种模式用于数据的输入、输出，包括数据的流式读写、直接HDFS读写等，可根据作业处理的数据量与集群机器硬盘容量，视情况决定所采用的读写方式。

## 可视化界面

为方便用户查看作业信息，XLearning提供可视化界面用于展示作业执行进度和输出日志等内容。作业执行完毕后，亦可查看日志内容，便于分析训练过程进展。对于TensorFlow类型作业，支持TensorBoard服务。作业运行界面大致分为三部分（如下图所示）：

- All Containers：显示当前作业所含Container列表及各Container对应信息，如Container ID、所在机器（Container Host）、所属类型（Container Role）、当前执行状态（Container Status）、开始时间（Start Time）、结束时间（Finish Time）、执行进度（Reporter Progress）；
- View TensorBoard  
：当作业类型为TensorFlow时，可点击该链接直接跳转至TensorBoard页面；
- Save Model  
：用户可在作业执行过程中，可以将当前训练模型的输出结果上传至HDFS，并显示目前已上传的模型列表。

**Tensorflow Application application\_1510915867109\_0012** Logged in as: dr.who

All Containers:

Container ID	Container Host	Container Role	Container Status	Start Time	Finish Time	Reporter Progress
container_e28_1510915867109_0012_01_000003	---	worker	RUNNING	Tue Nov 21 18:18:29 CST 2017	N/A	<div style="width: 100%;"></div>
container_e28_1510915867109_0012_01_000004	---	worker	RUNNING	Tue Nov 21 18:18:31 CST 2017	N/A	N/A
container_e28_1510915867109_0012_01_000002	---	ps	RUNNING	Tue Nov 21 18:18:31 CST 2017	N/A	N/A

View TensorBoard:

Tensorboard Info  
<http://---:32954>

Save Model

saving the model ... 0/2

Saved timeStamp	Saved path
2017-11-21 18:18:47	/tmp/tensorflow_model/interResult_2017_11_21_18_18_47

## 原生代码兼容

XLearning支持TensorFlow分布式模式的ClusterSpec自动分配构建，单机模式和其他深度学习框架代码不用做任何修改即可迁移到XLearning上，便于用户快速使用。

## Checkpoint功能

利用深度学习框架本身的Checkpoint机制和直接读写HDFS数据功能，XLearning方便用户实现训练恢复继续执行。

XLearning开源版本环境依赖简单，可直接运行于社区Hadoop版本，使用方便，入门学习成本极低。公司的Yarn版本是我们在社区版本上做了不少增强，比如支持GPU的资源调度、GPU通信亲和性的感知、DockerContainer支持等。依赖于这些特性公司在用的版本多出了GPU资源调度支

持、作业Docker化、临时GPU虚拟机、Container Metrics可视化图表展示等功能。这些功能我们后续会通过提供Yarn Patch或者开源自用Yarn版本来分享给大家，也欢迎大家随时跟我们沟通。

## 快速入门

在 XLearning 客户端，使用 \$XLEARNING\_HOME/bin/xl-submit 提交脚本将作业提交至Yarn集群。以 TensorFlow 作业提交为例：

### 上传训练数据至hdfs路径

将发布包解压后的data文件夹上传至hdfs，如：

```
cd $XLEARNING_HOME
hadoop fs -put data /tmp/
```

### 提交运行

```
cd $XLEARNING_HOME/examples/tensorflow
$XLEARNING_HOME/bin/xl-submit \W
--app-type "tensorflow" \W
--app-name "tf-demo" \W
--input /tmp/data/tensorflow#data \W
--output /tmp/tensorflow_model#model \W
--files demo.py,dataDeal.py \W
--launch-cmd "python demo.py --data_path=./data --save_path=./model --log_dir=./eventLog -
-training_epochs=10" \W
--worker-memory 10G \W
--worker-num 2 \W
--worker-cores 3 \W
--ps-memory 1G \W
--ps-num 1 \W
--ps-cores 2
```

提交脚本各参数含义如下：

参数名称	含义
app-name	作业名称为 "tf-demo"

参数名称	含义
app-type	作业类型为 "tensorflow"
input	输入文件，HDFS路径：/tmp/data/tensorflow，对应本地路径 ./data
output	输出文件，HDFS路径：/tmp/tensorflow_model，对应本地路径 ./model
files	需要传给各 container 的本地文件，包括 demo.py、dataDeal.py
launch-cmd	训练执行命令
worker-memory	worker内存使用为10G
worker-num	worker数目为2
worker-cores	worker使用CPU核数为3
ps-memory	parameterServer内存使用为1G
ps-num	parameterServer数目为1
ps-cores	parameterServer使用CPU核数为2

更多相关参数详细说明请见[运行提交参数](#)部分。

**本博客文章除特别声明，全部都是原创！**  
**原创文章版权归过往记忆大数据（过往记忆）所有，未经许可不得转载。**  
**本文链接：【】（）**