

12月23日，FATE开源社区第六期圆桌会圆满落幕。本次圆桌会，微众专家马国强为我们讲述了FATE 1.6版本的最新进展，并讲解了FATE 1.6版本的纵向算法优化方案，以及对FATE-Client和DSL V2的介绍说明。接下来我们看一下经典的问答环节，并对其一一解析。

**直接产生100万个RI，这个是一个种子产生的，会不会更不安全？**

FATE没有用特定的种子去生成伪随机数，我们使用的是系统随机数，在python里面这个操作时比较耗时的。

**请问使用PipeLine是不是就不用像现在一样写DSL配置文件了？**

是的，从1.5版本开始，DSL定位是用于系统交互的，而Pipeline主要是面向开发人员，提供易用的python编程接口去搭建建模的DAG流程。

**感觉那个预生成r，是不是不安全？**

理论上来说，预先生成部分随机数r，不同的样本id随机分配到同一个r，因为是系统随机数，所以无法确认是哪几个样本id被随机分配到同一个id上，安全性是比较高的，另外用户也可以设定随机数r的复用比例。

**DSL v2的host列表按照0 1 2编号拆出来，怎么对应host的party id list？**

对应的是DSL里面role的host列表的顺序来映射的。

**FATE现在横向和纵向都能支持各种网络吗？比如Bert Yolo这种？**

目前支持的是DNN和CNN，对RNN模型或更复杂的模型在后续版本中会陆续支持。

**Pipeline与Flow有哪些区别？功能是相互覆盖的吗？**

功能不重叠。Flow是FATE的联邦多方调度服务，而Pipeline则是用于搭建建模DAG流程图的python编程接口。

**Flowclient可以生成预测的DSL么？**

可以的，当前可以使用flow-client v2或者flow sdk、pipeline的相关指令接口去生成预测的DSL。

**新版本对DSL V1兼容吗？**

是完全兼容的。在新版本发布前，我们做过了充分的兼容性测试，但建议用户逐步迁移到DSL V2。

**刚说的算法优化是针对eggroll的，还是对spark也有效？**

1.6版本的纵向算法优化流程是对具体算法过程的优化，不针对特定的计算存储引擎。

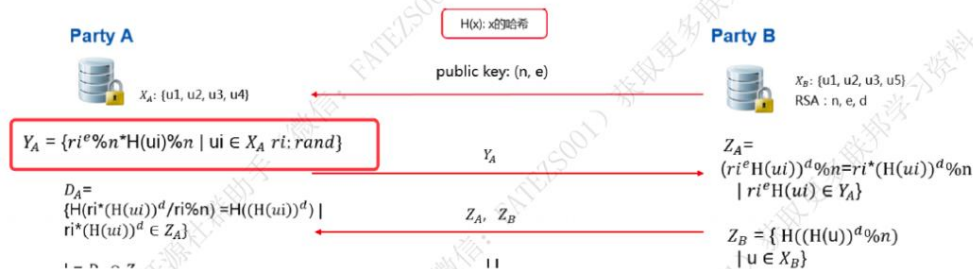
**如果A方做了优化，B方没做，性能会有提升吗？还是双方必须一致？**

需要具体算法具体分析，纵向LR主要耗时是在求梯度的加密数乘操作，优化单边价值不大的，但SBT来说耗时主要是host方的加密直方图统计，优化A单边可以获得很大的性能提升。

以下为本次圆桌会的部分内容介绍，添加小助手（FATEZS001）可获取详细资料：

# 01 FATE-1.6纵向算法优化

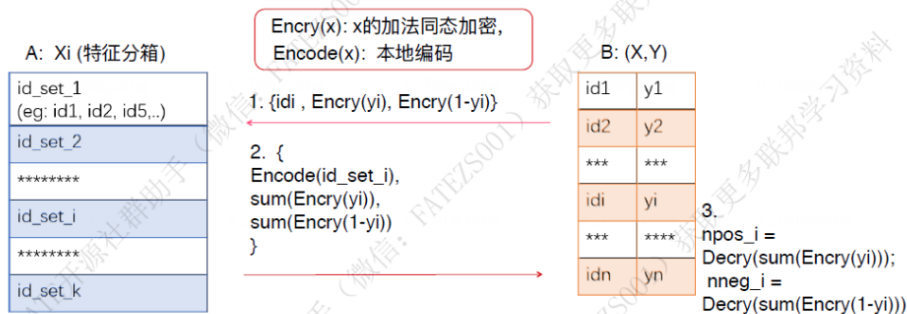
## 基于隐私保护的样本id匹配



### 性能优化1:

- $ri^e$ :  $Y_A$ 计算主要耗时为系统随机数生成和幂模运算，预先生成特定比例的 $ri^e$ ，样本随机复用

## 联邦特征工程



### B 方本地计算

1.  $dispos_i = npos_i / pos\_total$ ;  $distneg_i = nneg_i / neg\_total$
2.  $Woe_i = 100 * \log(dispos_i / distneg_i)$
3.  $IV = \sum_{i=1}^k (dispos_i - distneg_i) * \log(dispos_i / distneg_i)$

## 02 FATE-Client介绍

## CLI V2

- Enable
  - `pip install fate-client (pip install -e python/fate_client`
  - 初始化: `flow init --ip 127.0.0.1 --port 9380` (ip: flow服务端所在的主机地址, port: flow服务端口)
    - `flow -h`

```
(venv) [root@localhost ~]# fate flow -h
Usage: flow [OPTIONS] COMMAND [ARGS]...

Fate Flow Client

Options:
  -h, --help  show this message and exit.

Commands:
  component  Component Operations
  data       Data Operations
  init       Flow CLI Init Command
  job        Job Operations
  model      Model Operations
  queue      Queue Operations
  table      Table Operations
  tag        Tag Operations
  task       Task Operations

(venv) [root@localhost ~]# fate$ flow job
Usage: flow job [OPTIONS] COMMAND [ARGS]...

Provides numbers of job operational commands, including submit, stop, query and etc.
For more details, please check out the help text.

Options:
  -h, --help  Show this message and exit.

Commands:
  config  Config Job Command
  dsl     Generate Predict DSL Command
  list    List Job Command
  log     Log Job Command
  query   Query Job Command
  stop    Stop Job Command
  submit  Submit Job Command
  view    Query Job Data View Command
```

## PipeLine

```
train_pipeline = Pipeline.load_model_from_file("pipeline_saved.pkl")
train_pipeline.describe()
```

[illegible]

- Step8: 训练模型的恢复和查看
  - `Pipeline.load_model_from_file(restore_model_path)`: 恢复保存的pipeline
  - `pipeline.describe()`: 获取pipeline的相关描述, 包括阶段、创建时间、构造的dsl等

## 03 DSL V2使用介绍

### DSL V2

- 增加多项语法异常检测，V1兼容
- 优化提交任务配置格式
- 支持按需生成预测配置
- 预测阶段可实时编辑增加新组件

### DSL V2

- “component\_parameters”: 运行时组件的参数配置
  - “common”: 配置在下面的组件，代表所有方都是使用该参数设置
  - “role”: 指定某一方特定的组件运行参数
    - 图例 {“role”: {“host”: {“0”: {}}}}, host: “0” 代表 host 方的第0个party
    - 图例 {“role”: {“guest”: {“0”: {}}}}, guest: “0” 代表 guest 方的第0个party

```
"component_parameters": {
  "common": {
    "hetero_secure_boost_0": {
      "task_type": "classification",
      "num_trees": 3
    }
  },
  "role": {
    "guest": {
      "0": {
        "reader_0": {
          "table": {
            "name": "breast_hetero_guest",
            "namespace": "experiment"
          },
          "dataio_0": {
            "with_label": true
          }
        }
      }
    },
    "host": {
      "0": {
        "reader_0": {
          "table": {
            "name": "breast_hetero_host",
            "namespace": "experiment"
          },
          "dataio_0": {
            "with_label": false
          }
        }
      }
    }
  }
}
```

获取会议 PPT，或对圆桌会还有别的疑问？欢迎联系 FATE 开源社区助手获得帮助。

原文链接：<https://mp.weixin.qq.com/s/lrvKhTTLlz9SldzaogTXg>