

1月21日，FATE开源社区第七期圆桌会圆满落幕。本次圆桌会，微众银行高级研究员魏文斌为我们详细介绍了联邦学习的新领域——FedVision，并对其原理进行了相应的演示。接下来，我们先回顾下 FedVision 的基本信息，并对本次圆桌会问答环节中，社区朋友们提出的问题进行详细讲解。

### FedVision 介绍

FedVision 是首个轻量级、模型可复用、架构可扩展的视觉横向联邦开源框架，内置 PaddleFL/PaddleDetection 插件，支持多种常用的视觉检测模型，助力视觉联邦场景快速落地。FedVision 是基于 Python 实现的，作为一个视觉横向联邦方向机器学习项目，已经在 GitHub (<https://github.com/FederatedAI/FedVision>) 上开源首个版本 FedVision v0.1。

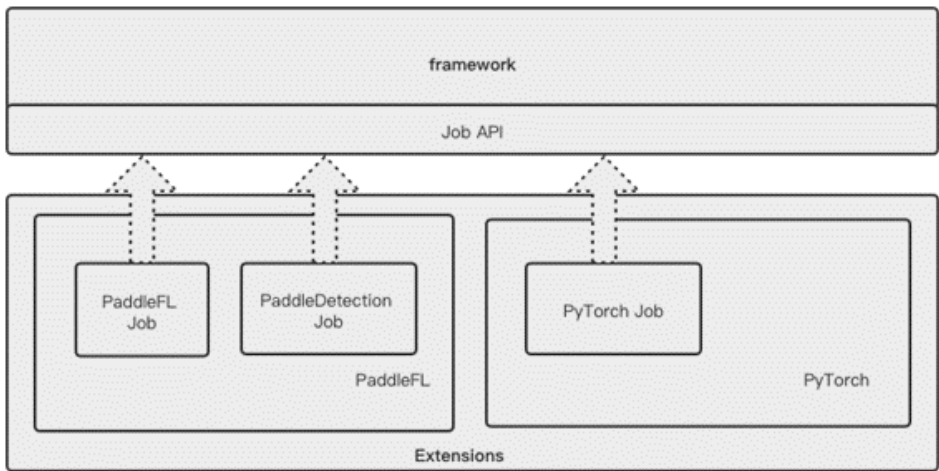
FedVision 项目的主要目标是：

- 🚀 在学术场景下，助力实验人员快速验证相关的实验想法。
- 🚀 在实际生产环境中，助力视觉横向联邦项目进行快速落地。

为了快速实现最小可用版本，FedVision v0.1 版本借助 PaddleFL 项目的部分能力，实现视觉领域的横向联邦建模功能。由于借助了 Paddle 的丰富生态，经简单的调制适配，FedVision v0.1 即可直接使用 PaddleDetection 项目实现的几乎全部的视觉检测模型。

### FedVision 逻辑构图

FedVision 从项目逻辑上可分成两个部分：



1. 框架

FedVision 的框架部分由几个相对独立的小组件构成，相互之间通过网络通信配合实现整个联邦学习任务的生命周期。这些组件根据所属的角色不同又归为两类：

🚦 Coordinator

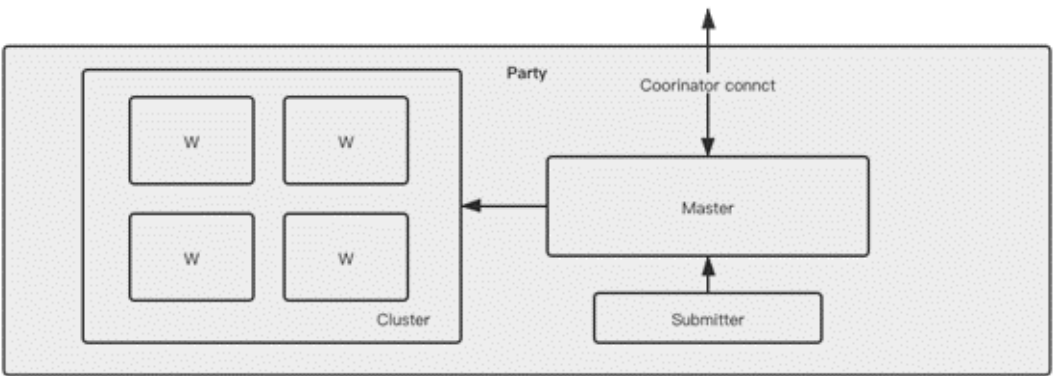
这个组件是单独的功能模块，在横向联邦学习任务的生命周期中属于协调者的角色，被所有的 “Party” 订阅。

当前版本所有的建模任务需要发布到 “Coordinator” ，由其分发给所有订阅对应类型任务的 “Party” 端。在所有收到订阅的参与方应答（或者超时）之后， 根据规则筛选出最终参与建模的所有 “Party” ，最后才分发任务的最终都任务配置。

目前版本的 “Coordinator” 只负责命令层面（任务信息）的信息分发，数据由 “Party” 任务之间直接建立的通信传递。在后续的版本中，数据的聚合分发功能将由 “Coordinator” 直接负责，从而推动 FedVision 的 “Party” 端往端上部署的能力。

🚦 Party

目前 Party 端包含了一个简单的 “Cluster” 组件，负责往多个不同的计算节点分发 Task、一个 “Submitter” 组件（供用户提交任务）以及一个 “Master” 组件（负责对话 “Coordinator” 和 “Submitter” ）。





## 2. 使用便捷

通过集成 PaddleDetection 的能力，在用户构建视觉检测模型的时候，简单配置一份 yaml 文件即可，不需要写一行代码！

对于想自定义模型的用户，框架也不限制您的发挥，类似于内置的卷积网络 demo 手动实现也相当容易。

## 3. 内置丰富模型

在现有的实际生产场景中，最适合横向联邦技术的场景的就是视觉检测相关的建模任务。

借助项目架构扩展的能力，目前主流的视觉检测模型均可使用，“Yolo”，“rcnn”一家子全都安排。

### 期待更多开发者加入

FedVision 将会持续完善，为开源社区提供更加友好强大的横向视觉联邦能力。我们将在下一版本支持 PyTorch 的扩展，虽然这依赖于 Coordinator 特性的升级。我们还将完善对加密聚合协议的支持，引入结构化压缩聚合等技术。另外，更轻量的客户端也在计划之中。

### 圆桌会问答环节

#### FedVision 跟 FATE 的关系是啥？看介绍用的是 PaddleFL？

FATE 目前已经具备横向联邦通用框架，在 FATE 可以实现纵向联邦和横向联邦一体化应用。FedVision 吸收了 FATE 项目上积累的横向联邦框架经验，同时又要更加聚焦视觉这个特定领域。所以，我们决定尝试用一种轻量化的框架去快速验证我们的一些想法。

引入 paddle 生态，可以复用 paddle 生态已有的成熟视觉功能比如 paddledetection, paddleocr 等，未来我们也会计划引入 pytorch 生态。FedVision 轻量化框架也更加适合未来边缘端部署和应用。可以说 FedVision 是横向视觉方向的先头部队，更适合探索一些新的概念，等成熟之后会反馈给 FATE 项目。

#### 视觉上的具体实现和应用场景可以详细讲解一下么？

今天本身来说是给大家介绍下 FedVision，因为 FedVision 对大家来说是比较陌生的，后面结合应用场景的话，看看有没有机会以技术博客的方式给大家介绍一下。

#### 对训练的权重进行聚合然后分发给客户端？

对。这个类似于数据并行的分布式训练，一般有两种模式：

- 各方用本地数据训练本地模型之后，对模型进行聚合。
- 各方用本地模型训练本地模型之后，计算出梯度（模型差值），对梯度进行聚合。

#### Party 中 user 和 client 的分工和协作分别是？

Party 中的 user 就是真实的人，就是去操作 client，通过一些简单的客户端命令进行交互，它们相当于是 user 提交任务给 client 这样的关系。

### **FedVision 的任务调度为什么不用 FATE 现有的任务调度呢？**

首先，如果是要用 FATE 调度的话，需要每方都部署一个 FATE，另外如果我们考虑往端上做的话这个东西有点不太现实，因为这样操作目前来看太重了。当然这不代表 FATE 以后做不了，只是按照项目的迭代周期，FATE 不会这么快做这件事情。所以我们开发一个 FedVision 项目来做一些横向联邦方向的快速迭代。

### **请问下 FedVision 现在都能支持哪些模型？**

基于 Paddle 来做的，准确来说利用了 PaddleDetection 的项目，所以基本上它支持的模型我们现在都能用，也就是基本上主流的视觉模型都可以。

### **医学的图像联邦能说下是都交互了哪些参数么？**

训练的算法用的 YOLOv3 的算法，参数是模型本身的参数。

### **FedVision Paper 的模型压缩等算法，也在开源的版本里面吗？**

模型压缩现在还没有做，后面会做这个东西。

### **本地训练可以多讲讲吗？如果本地没有训练所需要的资源要怎么解决呢？**

现在的版本还处理不了这个问题，但是这个要解决也很容易，因为 Coordinator 广播相关的任务信息后，client 方可以选择加入也可以选择加入，如果计算资源不够可以选择不加入。

### **训练过程中，参与方掉线后重新加入怎么处理？**

参与方掉线，可以自己再选择加入后续的任务等待，但是之前参与的训练就没法再加入了。

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：

## **大纲**

1. 横向联邦学习的基础知识
2. 简要介绍 FedVision
3. FedVision 使用的简单演示
4. FedVision 的后续规划



<https://www.fedai.org>

## 横向联邦训练的基本模式

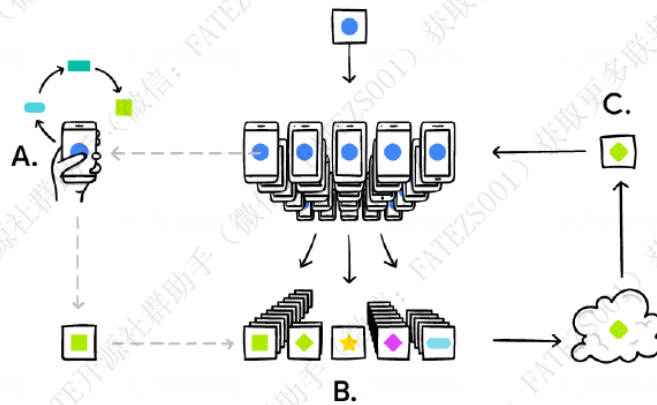


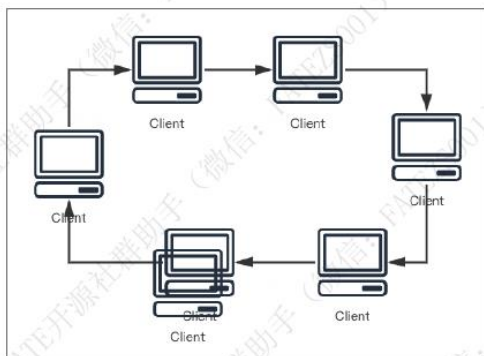
Figure 1: (A) users' updates are aggregated (B) to form a consensus change (C) to the shared model, after which the procedure is repeated.  
<https://www.fedai.org>

## 什么是 FEDVISION

- 横向联邦
- 视觉
- Python 实现
- 开源, FederatedAI, [GitHub: FederatedAI/FedVision](https://github.com/FederatedAI/FedVision)

<https://www.fedai.org>

"ring reduce"怎么样?



<https://www.fedai.org>

### 扩展

- 如何进行本地训练?
- 如何聚合模型参数或者梯度?
- 用户如何定义模型?



<https://www.fedai.org>

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/41TVzaG6oBLhNYbGy6qztzw>