关于项目的研发设想与计划

蔡剑平

通过对于项目研发的理解和项目相关技术的调研，本文从项目的工程设计的角度总结了项目研发的一些设想和计划。通过初期调研来看，选择 FATE 作为项目的开发框架在一定程度上是合理的。主要体现在以下方面。其一，采用 FATE 框架能够在技术层面上很好地体现“联邦学习”这一个主题；其二，FATE 作为一个目前相对成熟的联邦学习商用框架，已经具备了一系列任务管理、数据可视化的功能，对于后期成果保留，成果呈现具有显而易见的好处。其三，FATE 能够支持拓展到集群开发环境，从而实现与医院的联合应用示范。上述好处可以帮助我们在项目研发中节省大量工作量并且实现更加规范更加有效率的开发工作，因而选择 FATE 框架并不是一个错误的选择。

然而，基于框架的开发，尤其是 FATE 之类的企业型框架有些代价是需要付出的。首先是框架学习成本，对于任何一个框架来说，这都是需要的。对于 FATE 框架来说，从表层应用来理解框架的运作并不困难，通过网络上提供的文档或样例可以较快地上手。难的是如何理解 FATE 中所涉及的联邦算法协议，这对于后期的研发工作实际上是重要的。如果不能很好地理解，后期研发将可能产生难以解决的问题。因此，我认为每个需要关注的组件或算法需要经过数天时间了解算法及协议运行机制。其次，FATE 作为一套企业级框架，它包含了多个层次结构以及相互独立的前后端，这一点与 TensorFlow、Pytorch 等存在本质区别。这实际上给开发工作提出了挑战。第一个挑战是前后端分离的框架结构 FATE 框架提供的算法对于前端来说只是一个黑盒，开发者无法直接通过开发工具直接获得或者访问 FATE 各个组件算法的内部状态。其实，目前的 FATE 框架并不能支持局部的单元调试。虽然通过技术调研可以 FATE 提供了 unittest 模块，但经过尝试可以确定该模块并不能充分模拟真实的计算环境。换句话算法，对于算法的一点修改后的测试必须完整走一遍算法流程，这将导致项目研发和迭代消耗极大的时间成本。

在项目研发初期，正视这些挑战并设计合理的开发思路和开发模式至关重要。正所谓“工欲善其事必先利其器”，研发时第一需要解决的是开发环境的设计问题。这些问题包括如何实现与 Docker 内环境的协同开发、如何结合 FATE 框架的设计输出 FATE 算法的内部状态、如何通过模拟或部分模拟 FateFlow 运行环境来实现子功能的调试、如何针对开发需求寻找合适的工具来提升开发效率等。例如，在与 Docker 中 FATE 环境实现联合开发的过程中，直观可见或网上可查几种开发方式包括，利用 XShell 等工具直接在命令行中写代码、利用 Docker 和主机之间的文件映射通道间接地上传代码、利用 Pycharm 的 Docker 连接模块来关联远程 Docker 镜像以及放弃 Docker 而使用主机开发替代。面对这些问题，作为开发者需要一个一个尝试、评估，并选择一个最优的方案。而对于本项目来说，这些方法均无法满足高效开发的需求，正确的解决方案是在 Docker 中从零搭建一个 SSH 服务，并通过 SSH 服务实现远程开发。然而，该解决方案在网络上或者常规项目的开发中是没有的，需要结合对 Docker 技术和 FATE 框架的理解以及多次尝试才能实现。

不可否认的是，上述工作是开发者所应该独立完成和解决的。但这些工作琐碎、耗时且难以作为直接的绩效呈现。这也是为什么规划上看简单的几个步骤在实际操作时确需要漫长时间的重要原因。我认为，一个好的开发者不能单纯地面向项目的交付开发，还必须考虑项目的迭代、拓展、测试、开发效率等技术方面的因素。这些任务的工作量通常不是一个人全职开发可以快速搞定的，需要多个人协同配合才能保证以足够高的效率。因此，关于我们的医疗联邦学习项目，我认为在项目初期医疗数据可能不是最主要的，医疗数据的形式可能比医疗数据的内容更加重要。一方面，目前通过沟通了解到目前医院那边的数据进展还需要一

定时间，目前只能知道医疗数据包括了一些医疗影像、诊断信息以及多个分型，具体细节难以知晓；另一方面，医疗数据具有较高的保密程度，即使在后期医疗数据可能也难以提供。因此，在前期数据准备不足的情况下，我认为应当把研发的重点放在 FATE 框架的技术细节上，并在研发方式或技术探索上下更多的功夫，同时针对性地研究其中与后期开发相关的算法或组件（如 HeteroNN）。反而，一开始就花时间在用医疗数据给出一些初步可见的成果是不必要，这些完全可以用 Github 提供的样本数据来解决。只要把握数据形式，后期在将模型与医疗数据对接的时候做好一些数据匹配、转换、预处理工作即可。不过，我认为根据已知的一些需求针对性去寻找格式符合项目设计要求的数据是有必要的。因为符合特定联邦学习场景且公开的医疗数据在网上并不容易寻找，因此我认为当合适的医疗数据暂时未找到，寻找同类型的数据也是合理的。

关于项目开发方面，我的开发思想是一种“步步为营”的思想。项目开发前期，不应追求任何高大上的目标，而是把开发环境部署，开发技术的探索做到位。一开始应当从最简单的例子入手，一步一步地追踪并了解该例子做了什么，数据如何流通，结果如何，哪些东西是可变的。接下来，尝试做一些小的修改，测试每个值得关注的变量（包括数据）在算法模块中的作用以及哪些变量设置是冗余的（例如 HeteroNN 提供的例子中，对 Host 设置交互层就是一个无效的操作）。而后拓展到一些更加真实的场景，利用符合目标场景的数据运行一个最简单的模型以保证数据合法。再然后，尝试在非联邦学习的场景（非 FATE 框架）下利用目标数据实现模型并使得模型的实现接近于 FATE 环境的开发要求，从而使得实现的模型能够以尽可能少的修改移植到联邦学习场景（FATE）中。完成这个步骤后，就可以考虑引入更高级模型，迭代地开发，并精炼算法的效果。而到了后期，虽然我不太了解医院允许什么程度的数据接入，但可以肯定的是，没有直接应用医疗数据测试的模型是不可靠的，此时就应当根据与医院合作的情况设计相应的线上环境部署方案和项目测试方案。实际上，有企业开发经验的应该都明白，线上部署是一件极度消耗人力、时间的事情，必须多部门协调配合、熬夜加班才能完成。

最后，具体到我们的这个项目，我认为有一些挑战是接下来需要面临的。一个是数据，网上医疗数据不少，但是符合特定场景的联邦医疗数据不好找。我认为尝试跟医院沟通，让他们提供一些严格加密保护（如使用差分隐私）的小样本的数据集来支持项目开发是必要的，但这么做是不能保证所设计的模型在医疗数据上的效果的。第二是联邦神经网络在理论上我认为是完善的，但在初期研究中它的实际表现存在不稳定因素（如严重偏移的类别分割点，即使 AUC 值很高）。我认为这些不稳定因素可能斗志联邦学习后的效果好于单独一方的基本目标无法实现。还有，就是如何结合医院的数据进行协同开发，即如何在医院方要求数据高度保密的情况下进行算法研发和测试。

以上是我从一个项目设计者的角度出发提出的一些见解。这些见解是基于本人过往的项目经验和对相关从业人员的观察提出的，并不能保证科学性以及充分的合理性。上述观点和内容带有局部性，并不适用于任何人或任何情况。例如，本文对于非常熟悉 FATE 框架或能力非常强的开发者而言是不适用的。