**多方安全计算的技术介绍**

## 总体介绍



算法核心流程如上图所示，解释如下。

**Step 1：初始化模型**

Alice初始化前k个特征的模型WA，将WA随机切分, 然后将其中一份随机数发送给Bob；

Bob初始化后n-k个特征的模型WB，将WB随机切分, 然后将其中一份随机数发送给Bob；

**Step 2: 采用随机梯度下降，重复执行如下步骤，直至收敛**

**Step 2.1**：Alice和Bob拥有自己的特征（XA和XB），同时Alice和Bob分别拥有部分模型分片（随机数）。然后双方使用秘密分享矩阵乘法计算WX，其结果是Alice和Bob分别拥有计算结果的一个分片（随机数），即<WX>1和<WX>2;

**Step 2.2**：Alice和Bob拥有自己的特征（XA和XB），同时Alice和Bob分别拥有部分WX的分片（随机数）。然后双方使用秘密分享矩阵加法和乘法计算梯度g，其结果是Alice和Bob分别拥有各自模型分片相对比的梯度（也是随机数），即<gA>1，<gB>1和<gA>2，<gB>2；

**Step 2.3**：Alice和Bob分别更新各自的模型分片（更新之后也是随机数）

**Step 3：模型汇总**

Alice把迭代完成后的模型分片发送给Bob，Bob把迭代完成后的模型分片发送给Alice；

Alice汇总得到自己特征对应的模型WA， Bob汇总得到自己特征对应的模型WB。

注：以上过程，在Step2.2中，用到了泰勒展开，即将Sigmoid函数通过泰勒展开转换成线性计算。

## 安全性分析和证明

**秘密分享技术简介：**上述算法主要技术依赖于Shamir等人在1979年所提出的秘密分享技术。秘密分享技术是多方安全计算领域中的一项核心技术，被广泛应用于隐私保护机器学习算法中。其基本思想是把一个秘密切分成n份并发送给n个人，每个人保存了整个秘密中的一份share。这些参与者中的一个或者几个人无法窥探到秘密，而当真正要重构这个秘密时，需要拿到share的这n个人共同参与，这叫做 (n, n)秘密分享。在我们联合建模场景下，为两方的秘密分享。秘密分享的加法由参与方各自完成，非常简单；而秘密分享的乘法技术则依赖于Beaver在1991年所提出的Beaver’s Triple技术。需要注意的是，为确保随机数的产生可以服从均匀分布，我们实现的秘密分享技术都是在**有限域内**完成的。

**安全性分析**：上述算法双方的输入是各自的数据（包括特征及标签），输出是各自获得的模型，整个过程中不涉及双方隐私数据的直接交互。下面，我们来分析一下上述算法双方都涉及到哪些信息交互：

* 在Step 1中，双方各自随机初始化模型，并把各自的WA和WB秘密分享给对方，该过程等价于双方各自随机生成子模型<WA>1等，因此没有信息泄露；
* 在Step 2.1中，双方使用Beaver’s Triple技术去做安全的矩阵乘法，其过程交互的也是随机数，双方得到的结果也是随机数，因此没有信息泄露；
* 在Step 2.2中，双方再次使用Beaver’s Triple技术去做安全的矩阵乘法，其过程交互的也是随机数，双方得到的结果也是随机数，因此没有信息泄露；
* 在Step 2.3中，双方本地分别更新各自的模型分片，没有交互；
* 在Step 3中，双方将各自的模型分片（随机数）给到对方，用于恢复出明文的模型，因此没有信息泄露。

通过以上分析，我们可以发现，双方的隐私数据（包括特征X和标签Y）都一直由各自保留，而交互的信息只有随机数，因此，整个过程没有任何信息泄露。可以得到以下结论：**A方反推不出B方拥有的特征（XB）和标签（Y）；而B方反推不出A方拥有的特征（XA）**。

**安全性证明**：上述算法的严格性证明可以通过该安全性通过现实世界（real-world）和理想世界（ideal-world）的模拟范式来证明。证明的核心思路是分析中间交互的信息（如A方发送给B方的数据）是否不可区别的。在上述算法中，由于双方的交互信息都是有限域内的随机数，而随机数之间是无法区分的，因此是可证安全的。