基于知识蒸馏的联邦相互学习

2021 年 1 月 12 日

背景

Algorithm 1: 朴素联邦学习算法

Input: 带有各自不同数据的用户群体 $c_i \in C$，服务器 S，一个打算共同训练的模型 M

Result: 最终训练好的模型 M'

在服务器 S 上初始化模型 M;

while 模型达到某个精确度 **do**

  各个用户将服务器上的模型 M^t 下载到它们本地;

  各个用户利用各自的数据训练模型，更新模型，$m_i^t \leftarrow train_i(M^t)$;

  各个用户将各自的 m_i^t 上传到服务器 S 上;

  服务器将模型的参数平均，即 $M^{t+1} \leftarrow \frac{1}{n}\sum_n m_i^t$;

end

问题

1. 如果用户数据分布不均匀，训练效果不好，收敛过程复杂
2. 这种情况下最后只能有一个训练好的模型，用户和用户之间的模型个性化不够

解决方案

Algorithm 2: 基于知识蒸馏的联邦相互学习

Input: 带有各自不同数据 d_i 和不同模型 m_i ，但是训练目标一致的用户群体 $c_i \in C$，服务器 S，一个用作共同训练媒介的模型 M，一个先验模型 N，用户各自贡献出的公共数据集 $Dataset_{pub}$（不需要 label）

Result: 用户各自最终训练好的模型 m_i'

在服务器 S 上初始化模型 M, N;

将用户各自贡献出的公共数据集的一部分 $Dataset_{pub}^{part}$ 送入 N 中，获得软标签;

将 $Dataset_{pub}^{part}$ 和生成的软标签分给每个用户，扩充数据集;

while 各个用户的模型达到某个精确度 **do**

  各个用户将服务器上的模型 M^t 下载到本地 M_i^t;

  将公共数据集的剩余部分 $Dataset_{pub}^{res}$ 中一部分用作 M_i^t 和 m_i^t 的前向传播，获得结果 res_M 和 res_m;

  相互计算 res_M 和 res_m 的 KL 散度，得到 $L_{KL,M\to m}$ 和 $L_{KL,m\to M}$;

  各个用户利用各自的数据训练模型 m_i^t，得到损失 $L_{m,ori}$;

  根据以上损失反向传播和更新参数：$M_i^{t+1} \leftarrow Update(M_i^t, L_{KL,M\to m})$;
  $m_i^{t+1} \leftarrow Update(m_i^t, L_{ori} + L_{KL,m\to M})$;

  各个用户将各自的 M_i^{t+1} 上传到服务器 S 上;

  服务器将模型的参数平均，即 $M^{t+1} \leftarrow \frac{1}{n}\sum_n M_i^{t+1}$;

end

效果说明

测试数据集：Mnist，Cifar10，Fashion Mnist

- 收敛加速：同朴素的联邦学习相比，收敛时精度变化不大，但是收敛稳定性增加
- 用户的模型可以不相同：同 FedAmp 相比，训练更快速计算量更小