SAFA: A Semi-Asynchronous Protocol for Fast Federated Learning With Low Overhead

INTRODUCTION&RELATED WORK

联邦学习的特点

相比DML，FL考虑了数据隐私安全，数据不可移动

DML往往要求worker的性能是差不多的，一般是同构的，而FL天然是异构的

FL的终端设备和网络的不可靠性更高，可能会随时下线

不同worker的数据量不同，数据分布也不同

> DML的很多假设对于FL是不现实的，如同质的worker node，同构的高速网络

联邦学习同步算法的局限性(FedAvg)

随机选择一部分参与者的是不可靠的

通信效率低下,straggler问题

client利用率低，每一轮中除了被选为参与者以为的client都处于空闲的状态

进度被浪费

THE SAFA PROTOCOL

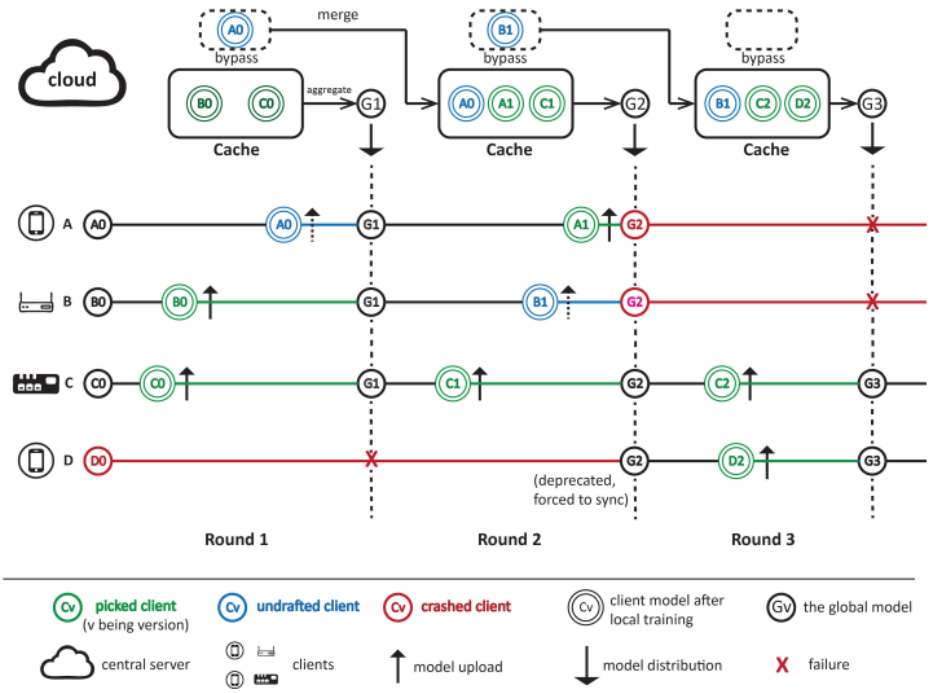

Fig. 1. The diagram of SAFA protocol showing the interaction between the cloud and end clients in different states

在一轮训练开始之前，SAFA将clients 分为三种状态

Up-to-date clients:完成了上一轮的local training，并且成功提交到server

Deprecated clients:仍在基于过于陈旧的global model进行local training

Tolerable clients:既不是基于最新的global model，但也不是过于陈旧的global model进行local training

> SAFA仅要求Up-to-date和Deprecated的client和server同步最新的global model，Tolerable clients则与server保持异步

完成一轮训练后，SAFA还会基于client selection的结果对client打一个标签

picked：client 本地训练的结果将被在本轮被采用

undrafted:client 本地训练的结果暂时不会被这一轮的aggregation采用，但是会被cache来供将来使用

crashed:client 因网络或主动退出等原因没有成功完成本地训练

lag tolerance:延迟容忍，SAFA中唯一的超参数

> The hyper-parameter lag tolerance in some ways controls the tradeoff between communication overhead and the convergence rate of federated optimization. If it is set too small, the server may suffer heavy downlink transmission as the portion of deprecated clients increases. If it is set too large, the convergence of the global model could be unsteady.

Client Selection

不同于FedAvg使用一个参数C来硬限制每轮运行参与的client的比例，在client愿意的情况下，SAFA允许所有的client参与，收到C 部分的updates后就可以结束一轮

> In our approach, we retain this hyper parameter but no longer apply it as a hard constraint. Instead, we release the restriction to allow all clients to participate if they are willing to, and enable the central server to end a round once C-fraction of updates have been received.

client selection的时机：post-training selection

选择参与的client 和实际完成了本地训练并且commit了update的client不是等价的，因此SAFA定义了EUR(Effective Update Ratio)来衡量有效更新的比例

$$EUR = \frac{|P - P \cap K|}{|M|}$$

P（Picked）K(Crashed)

EUR正比于P，反比于K，K不可控，而单纯提高P又是不可行的

使用post-training selection可以获得较好的EUR

> selection-ahead-of-training:先选C个client进行本地训练，然后等待C个client完成本地训练
>
> selection-after-training:允许所有的client进行本地训练完成，选择最快完成的C个client的结果

client selection 的策略:

principle:higher priority is given to those clients that are less involved

Algorithm 1: Compensatory First-Come-First-Merge (CFCFM) client selection

Input : round number t, client set M, last-round picked clients $P(t-1)$, selecting fraction C, round deadline T_{lim}

Output: clients to pick $P(t)$

$P(t) = \varnothing$
$Q(t) = \varnothing$
$quota = C \cdot |M|$
while $|P(t)| < quota$ and $T_{round} < T_{lim}$ **do**
    Await new updates
    $w'_k \leftarrow$ update arrives from client k
    if k not in $P(t-1)$ **then**
        add k to $P(t)$
    else
        add k to $Q(t)$
    end
end
if $|P(t)| < quota$ **then**
    Sort $Q(t)$ by arrival time
    $q \leftarrow quota - |P(t)|$
    $P'(t) \leftarrow$ first q clients in $Q(t)$
    $Q(t) \leftarrow Q(t) - P'(t)$
    $P(t) \leftarrow P(t) + P'(t)$
end
return $P(t)$

Discriminative Aggregation

Algorithm 2: Semi-Asynchronous Federated Averaging (SAFA) protocol

Input : maximum number of rounds r, client set M, local mini-batch size B, number of local epochs E, learning rate η, lag tolerance τ

Output: finalized global model

Server process: // running on the central server

Initializes client connections

Initializes global model $w(0)$ and the cache

for round $t = 1$ to r **do**

  Distributes $w(t-1)$ according to Eq. (3) given τ

  for each client k in M **in parallel do**

    $w'_k(t) =$ **client_update**$(k, w_k(t))$

  end

  Collects and selects client updates using CFCFM

  Updates cache according to Eq. (6)

  Performs aggregation and get $w(t)$ using Eq. (7)

  Updates cache according to Eq. (8)

end

return $w(r)$

Client process: // running on the client k

client_update(k, w_k)**:**

$B_k \leftarrow$ batches from D_k of size B

for epoch $e = 1$ to E **do**

  for batch b in B_k **do**

    $w_k = w_k - \eta \nabla f(w_k; b)$

  end

end

$w'_k = w_k$

return w'_k to the server

lag-tolerant distribution

$$w_k(t) = \begin{cases} w(t-1) & \text{if } k \in \bigcup_{v=t-1} M_v, \text{or } k \in \bigcup_{v<t-\tau} M_v, \\ & \text{// up-to-date or deprecated clients} \\ w_k(t-1) & \text{if } k \in \bigcup_{t-\tau \le v < t-1} M_v \\ & \text{// tolerable clients} \end{cases} \tag{3}$$

Aggregation Steps

(1) *Pre-aggregation Cache Update*:

$$w_k^*(t) = \begin{cases} w_k'(t) & \text{if } k \in P(t), \\ w(t-1) & \text{if } k \in \bigcup_{v<t-\tau} M_v(t), \\ w_k^*(t-1) & \text{otherwise} \end{cases} \tag{6}$$

where $w_k^*(t)$ denotes the k-th entry of the cache structure (see Fig. 1), and $w_k'(t)$ denotes the trained local model at round t. Entries of deprecated clients will be replaced with the global model $w(t-1)$.

(2) *SAFA Aggregation*:

$$w(t) = \sum_{k=1}^{m} \frac{n_k}{n} w_k^*(t) \tag{7}$$

(3) *Post-aggregation Cache Update*:

$$w_k^*(t+1) = \begin{cases} w_k'(t) & \text{if } k \in Q(t), \\ w_k^*(t) & \text{otherwise} \end{cases} \tag{8}$$

EXPERIMENTAL EVALUATION

metrics:

C：selection fraction

τ：lag tolerance

cr： client crash probability

EUR: Effective Update Ratio

SR：Synchronization Ratio

$$SR_{SAFA} = \frac{1}{rm} \sum_{t=1}^{r} (|\bigcup_{v=t-1} M_v(t)| + |\bigcup_{v<t-\tau} M_v(t)|) \tag{9}$$

> SR measures the usage of downlink by which the global model is distributed to the edge of network

VV：V V is defined based on the version distribution of local updates

$$VV_{SAFA} = \frac{1}{r} \sum_{t=1}^{r} \text{var}(V_t) \tag{10}$$

average round length:

$$T = \min\left\{T_{lim},\ T_{dist} + \max_k\{T_k^{down} + T_k^{up} + T_k^{train}\}\right\} \quad (17)$$

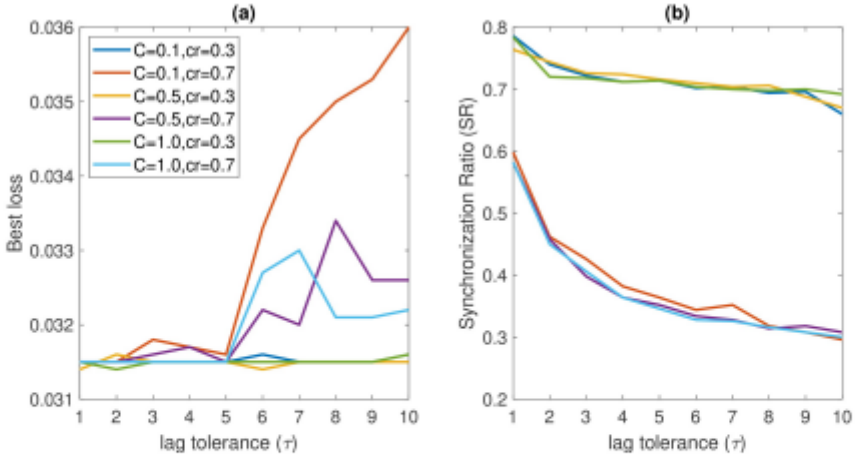

Fig. 3. (a) Best loss achieved by the global model and (b) the synchronization ratio over the federated optimization with SAFA protocol under different lag tolerance settings.

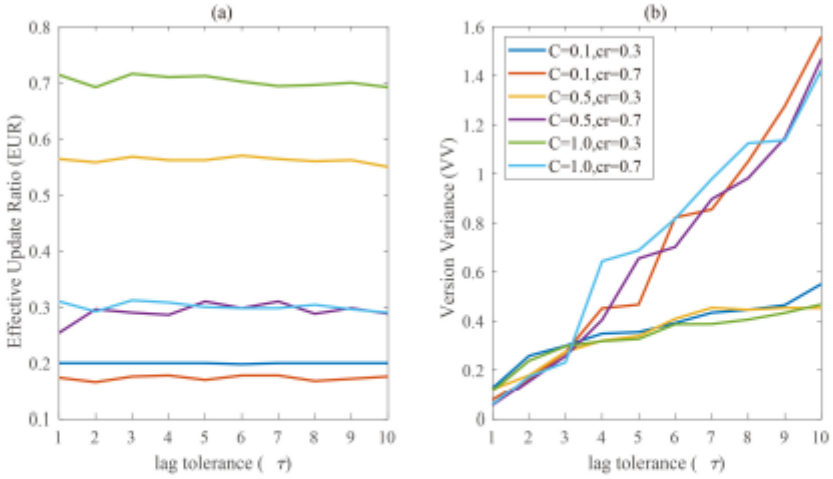

Fig. 4. (a) Effective Update Ratio (EUR) and (b) Version Variance (VV) over the federated optimization with SAFA protocol under different lag tolerance settings.

global model loss VS crash probability

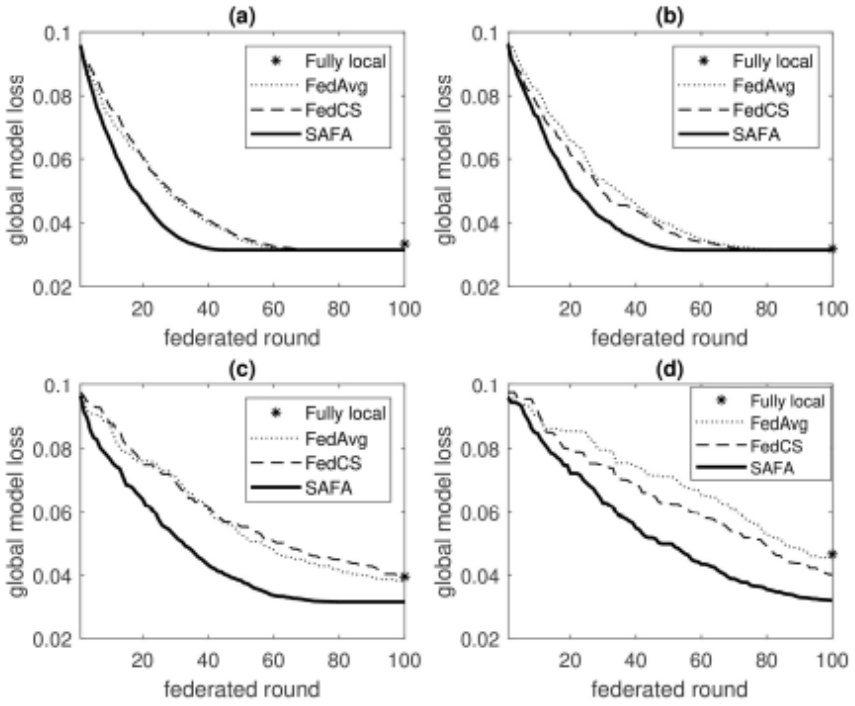

Fig. 6. The loss trace of the global model as the FL process progresses on Task 1 where the client fraction is set to 0.3 and the crash probability is set to 0.1, 0.3 ,0.5 and 0.7 for the four sub-figures (a)-(d), respectively

average round length VS client selection VS crash probability

Avg. round length (Task 1: regression)					
FedAvg					
cr	$C = 0.1$	$C = 0.3$	$C = 0.5$	$C = 0.7$	$C = 1.0$
0.1	316.22	489.37	586.90	731.12	808.59
0.3	429.63	652.39	641.40	736.53	832.02
0.5	372.43	495.37	475.14	621.91	676.41
0.7	354.34	405.86	593.10	728.25	661.67
FedCS					
cr	$C = 0.1$	$C = 0.3$	$C = 0.5$	$C = 0.7$	$C = 1.0$
0.1	207.50	487.47	564.20	656.49	786.96
0.3	336.97	519.58	651.23	401.95	832.02
0.5	186.51	221.46	467.98	621.91	676.41
0.7	195.09	398.81	584.68	393.09	661.67
SAFA					
cr	$C = 0.1$	$C = 0.3$	$C = 0.5$	$C = 0.7$	$C = 1.0$
0.1	149.69	389.44	540.41	606.48	734.40
0.3	202.44	430.68	583.22	371.77	699.23
0.5	169.33	215.66	408.85	510.85	508.23
0.7	161.81	293.09	402.18	411.06	379.29

思考

模型的可扩展性

server收到C-fraction的update之后，clients已经发出去了，造成带宽浪费