An Efficiency-boosting Client Selection Scheme for Federated Learning with Fairness Guarantee

论文要解决问题：在FL Client Selection中将公平性纳入考虑, 同时能够保证训练效率和训练效果

Motivation

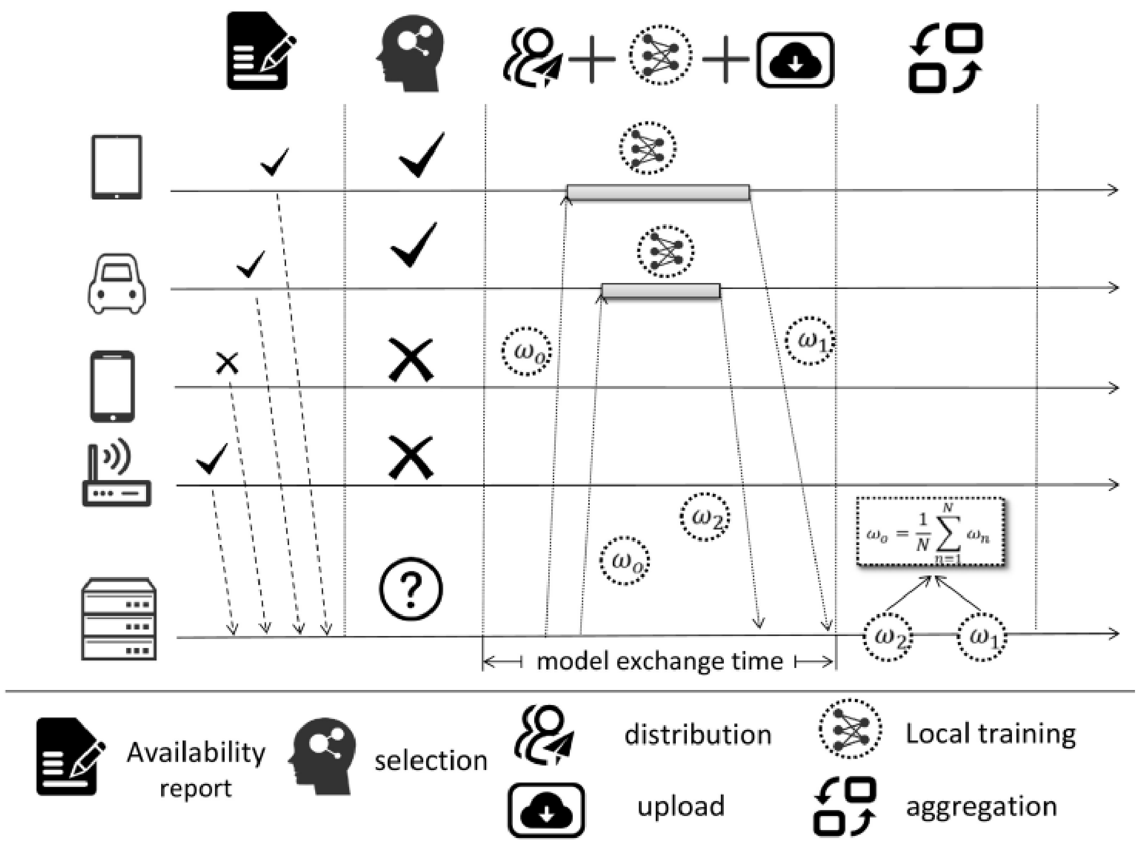
> From the perspective of a model owner, the selection decision in each round could have a profound impact on the model's training time, convergence speed, training stability, as well as the final achieved accuracy

作者认为之前的工作主要有俩个问题:

- 都假设了可以提前知道本地训练时间
- 更倾向于选择性能高的设备

> 总是选择快的设备可以加速训练速度，但是低速设备中的数据总是无法参与计算，因此作者认为是一种不公平的选择策略

System Model



1. client主动报告是否愿意参与这一轮训练，同时报告client端的信息
2. 调度器根据client报告的结果，进行client selection
3. 将global model分发到被选择的client，然后完成本地训练，并将本地模型上传
4. 汇聚上传的本地模型

论文聚焦解决client selection问题，目标是使得选择的client能使long-term average model exchange time最小，同时满足公平性等约束条件

第t轮model exchange time.

$$f(\mathcal{S}_t, \boldsymbol{\tau}_t) = \max_{n \in \mathcal{S}_t}\{\tau_{t,n}\} \tag{1}$$

S_t:round t被选择的client集合

$\tau_{t,n}$:round t client n的model exchange time

显然每一轮的model exchange time 由最慢的client决定

> 如果server能提前知道选择的client的model exchange time，那么每轮都总是选m个model exchange time最小的，那么long-term average model exchange time也是最小的，但这个策略不够公平，可能会影响模型的泛化能力

公平性约束

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[x_{t,n}] \geq \beta \quad \forall n \in \mathcal{N} \tag{2}$$

β models the expected guaranteed chosen rate of clients.

$x_{t,n}$is used to indicate whether client n is involved in the federated round t or not. In other words, $x_{t,n}$= 1 for n ∈ S_t; otherwise, $x_{t,n}$ = 0.

$$I_{t,n} = 1 \quad \forall n \in \mathcal{S}_t \tag{3}$$

$$|\mathcal{S}_t| = \min\left\{m, \sum_{n\in\mathcal{N}} I_{t,n}\right\} \tag{4}$$

An Offline Long-Term Optimization Problem

$$(P1): \min_{\{\mathcal{S}_1,\mathcal{S}_2,\ldots,\mathcal{S}_\infty\}} \quad \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T} f(\mathcal{S}_t, \boldsymbol{\tau}_t) \tag{5}$$
$$\text{s.t. } (2), (3), (4)$$

这个问题很难，甚至是不可能通过离线的算法方式解决的。作者通过引入Lyapunov optimization framework将该离线问题P1转换为一个在线问题。

1）为每个client引入一个虚拟队列，将上述公平性问题转为队列稳定性问题

$$Z_{t+1,n} = [Z_{t,n} + \beta - x_{t,n}]^+ \tag{6}$$

$Z_{t,n}$ 表示队列长度，β表示公式(2)中expected guaranteed selection rate,$[...]^+$表示max(...,0)

> 文章证明了保证公平性约束条件（2）和让队列保持mean rate stable是等价的

2）用Lyapunov optimization framework框架求解

定义Lyapunov function来描述在时刻t,虚拟队列积压的平方

$$\mathcal{L}(\boldsymbol{\Theta}(t)) = \frac{1}{2} \sum_{n \in \mathcal{N}} Z_{t,n}^2 \tag{8}$$

定义Lyapunov drift描述队列积压的增长

$$\Delta(\boldsymbol{\Theta}(t)) = \mathbb{E}[\mathcal{L}(\boldsymbol{\Theta}(t+1)) - \mathcal{L}(\boldsymbol{\Theta}(t)) | \boldsymbol{\Theta}(t)] \tag{9}$$

> 如果可以让增长存在上界，队列就可以mean rate stable

结合 objective function和Lyapunov drift为drift-plus-cost function

$$\Delta(\boldsymbol{\Theta}(t)) + V\mathbb{E}[f(\mathcal{S}_t, \boldsymbol{\tau}_t) | \boldsymbol{\Theta}(t)] \tag{10}$$

V是一个非负惩罚因子，用来做队列稳定和优化目标的tradeoff

论文证明drift-plus-cost function（10）存在上界

$$\begin{aligned} &\Delta(\boldsymbol{\Theta}(t)) + V\mathbb{E}[f(\mathcal{S}_t, \boldsymbol{\tau}_t) | \boldsymbol{\Theta}(\boldsymbol{t})] \\ &\leq \Gamma + \sum_{n \in \mathcal{N}} Z_{t,n} \mathbb{E}[\beta - x_{t,n} | \boldsymbol{\Theta}(t)] + V\mathbb{E}[f(\mathcal{S}_t, \boldsymbol{\tau}_t) | \boldsymbol{\Theta}(t))] \end{aligned} \tag{11}$$

因此，最小化drift-plus-cost function可以转化为最小化上述不等式的右边，问题P1也转化为P2

$$\begin{aligned} (P2): \min_{\boldsymbol{x}_t} \quad & \Gamma + \sum_{n \in \mathcal{N}} Z_{t,n}(\beta - x_{t,n}) + V\dot{f}(\boldsymbol{x}_t, \boldsymbol{\tau}_t) \\ s.t. \quad & \sum_{n \in \mathcal{N}} x_{t,n} = \min\left\{m, \sum_{n \in \mathcal{N}} I_{t,n}\right\} \\ & x_{t,n} \leq I_{t,n} \\ & x_{t,n} \in \{0, 1\} \end{aligned} \tag{12}$$

去掉P2中的常数，问题进一步简化为

$$
\begin{aligned}
(P3): \min_{\mathbf{x}_t} \quad & V \max_{n\in\mathcal{N}}\{x_{t,n}\tau_{t,n}\} - \sum_{n\in\mathcal{N}} Z_{t,n}x_{t,n} \\
s.t. \quad & \sum_{n\in\mathcal{N}} x_{t,n} = \min\left\{m, \sum_{n\in\mathcal{N}} I_{t,n}\right\} \\
& x_{t,n} \le I_{t,n} \\
& x_{t,n} \in \{0,1\}
\end{aligned} \tag{13}
$$

但P3目前还是不可解的，因为，每一轮的model exchange time在进行client selection的时候仍然是未知的，论文通过引入C^2MAB模型来估计model exchange time

$$
\tau_{t,n} = \boldsymbol{c}_{t,n}^{\top}\boldsymbol{\theta}_n^* + \epsilon_{t,n} \tag{14}
$$

$c_{t,n} \triangleq [1/\mu_{t,n}, s_{t,n}, M/B_{t,n}]^T$

$\theta_n^* \triangleq [\tau_n^b, \tau_n^s, 1/\eta]$

$\mu_{t,n}$指client n在round t可用的CPU率，$\mu_{t,n} = 200\%$表示2个CPU可用

τ_n^b表示如果用一个CPU完成本地训练需要的时间

$s_{t,n}$表示上一轮该client是否参与了训练，是为1，否则为0

τ_n^s表示client的冷启动时间

> 冷启动时间指数据准备时间，如将数据加载进内存的时间

M指模型size，$B_{t,n}$指带宽,$\eta \triangleq \log(1+SNR)$(SNR即信噪比)

含义其实就是model exchange time= 本地计算时间+冷启动时间+上传时间

$c_{t,n}$是一个动态，且是可以被scheduler提前知道的量，但θ_n^*是一个静态不变但不易获取的量，因此无法直接计算$\tau_{t,n}$,论文通过利用历史数据，采用岭回归估计出θ_n^*.

经过上述一系列转换，问题最终定义为P4，一个可解的整数线性规划问题

$$
\begin{aligned}
(P4): \min_{\mathbf{x}_t} \quad & V \max_{n\in\mathcal{N}}\{x_{t,n}\bar{\tau}_{t,n}\} - \sum_{n\in\mathcal{N}} Z_{t,n}x_{t,n} \\
s.t. \quad & \sum_{n\in\mathcal{N}} x_{t,n} = \min\left\{m, \sum_{n\in\mathcal{N}} I_{t,n}\right\} \\
& x_{t,n} \le I_{t,n} \\
& x_{t,n} \in \{0,1\}
\end{aligned} \tag{21}
$$

algorithm

注意到P4中第一项仅有有限的可能解，因此可以直接遍历这些可能的解。基于此，论文将P4划分成更小的子问题

$$
\begin{aligned}
\textit{(P4-SUB)} : \min_{\mathbf{x}_t} \quad & -\sum_{n \in \mathcal{N}} Z_{t,n} x_{t,n} \\
s.t. \quad & \sum_{n \in \mathcal{N}} x_{t,n} = \min \left\{ m, \sum_{n \in \mathcal{N}} I_{t,n} \right\} \\
& x_{t,n} \bar{\tau}_{t,n} \leq \bar{\tau}_{max} \\
& x_{t,n} \leq I_{t,n} \\
& x_{t,n} \in \{0, 1\}
\end{aligned}
\tag{22}
$$

论文中提出的算法如下

Algorithm 1 Divide-and-conquer solution for *P4*

Input:

The estimated time for model exchange; $\{\bar{\tau}_{t,n}\}_{n\in\mathcal{N}}$
The expected number of chosen arms; m
Indicator function of arms' availability; $\{I_{t,n}\}_{n\in\mathcal{N}}$
Length of virtual queue; $\{Z_{t,n}\}_{n\in\mathcal{N}}$

Output:

The solution for *P4* in round t; $\{x_{t,n}\}_{n\in\mathcal{N}}$

1: Set $\boldsymbol{Z}_t^* = \{Z_{t,n}\}_{I_{t,n}=1}$
2: Use $\mathcal{A}_t$ to store arms with an descending order of $\boldsymbol{Z}_t^*$
3: Use $\mathcal{N}_t^+$ to store all the n that satisfies $I_{t,n} = 1$
4: Set $k = \min\{m, \sum_{n\in\mathcal{N}} I_{t,n}\}$ // # of clients to be picked

5: **for** $n_{max} \in \mathcal{N}_t^+$ **do**
6:     Initialize an empty set $\mathcal{S}_{n_{max}}$
7:     **for** $n \in \mathcal{A}_t$ **do**
8:         **if** $\bar{\tau}_{t,n} \le \bar{\tau}_{t,n_{max}}$ **then**
9:             Push n into $\mathcal{S}_{n_{max}}$
10:         **end if**
11:         **if** $length(\mathcal{S}_{n_{max}}) == k$ **then**
12:             Calculate the objective of *P4* as $F_{n_{max}}$ based on $\mathcal{S}_{n_{max}}$
13:             Break the first loop
14:         **end if**
15:     **end for**
16: **end for**
17: Set n^* the index of minimum $F_{n_{max}}$ among those being calculated in line 12.
18: Return $\{x_{t,n}\}$ that represented by $\mathcal{S}_{n^*}$

完整的FL算法

Algorithm 2 Reputation Based Client Selection with Fairness (RBCS-F)

Input:

The expected number of involved clients each round; m
Exploration parameter; $\alpha_0, \alpha_1, \ldots$
The set of clients; $\mathcal{N}$, Parameter for ridge regression; λ
The guaranteed participating rate; β
Parameter for objective balance; V

Output:

The control policy $\pi = \{x_{t,n}\}_{n\in\mathcal{N},t=0,1,\ldots}$

for $n \in \mathcal{N}$ **do**
  Initialize $\mathbf{H}_{0,n} \leftarrow \lambda\mathbf{I}_{3\times3}, \mathbf{b}_{0,n} \leftarrow \mathbf{0}_3^\top, Z_{0,n} \leftarrow 0$
end for
for $t = 1, 2\ldots$ **do**
  Observe current contexts $\{\mathbf{c}_{t,n}\}$ and arms availability $\{I_{t,n}\}$
  for $n \in \mathcal{N}$ **do**
    $\hat{\boldsymbol{\theta}}_{t,n} \leftarrow \mathbf{H}_{t-1,n}^{-1}\mathbf{b}_{t-1,n}$
    $\hat{\tau}_{t,n} \leftarrow \mathbf{c}_{t,n}^\top\hat{\boldsymbol{\theta}}_{t,n}$
    $\bar{\tau}_{t,n} \leftarrow \hat{\tau}_{t,n} - \alpha_t\sqrt{\mathbf{c}_{t,n}^\top\mathbf{H}_{t-1,n}^{-1}\mathbf{c}_{t,n}}$
  end for
  // Execute Algorithm 1 for a decision
  $\{x_{t,n}\} \leftarrow$ Algorithm 1$(\{\bar{\tau}_{t,n}\}, m, \{I_{t,n}\}, \{Z_{t,n}\})$
  Distribute model to the selected clients and observe their model exchange time;$\{\tau_{t,n}\}$
  for $n \in \mathcal{N}$ **do**
    Update $Z_{t,n}$ according to (6)
    $\mathbf{H}_{t,n} \leftarrow \mathbf{H}_{t-1,n} + x_{t,n}\mathbf{c}_{t,n}\mathbf{c}_{t,n}^\top$
    $\mathbf{b}_{t,n} \leftarrow \mathbf{b}_{t-1,n} + x_{t,n}\tau_{t,n}\mathbf{c}_{t,n}$
  end for
end for

experiment

The queue length vs The penalty factor V

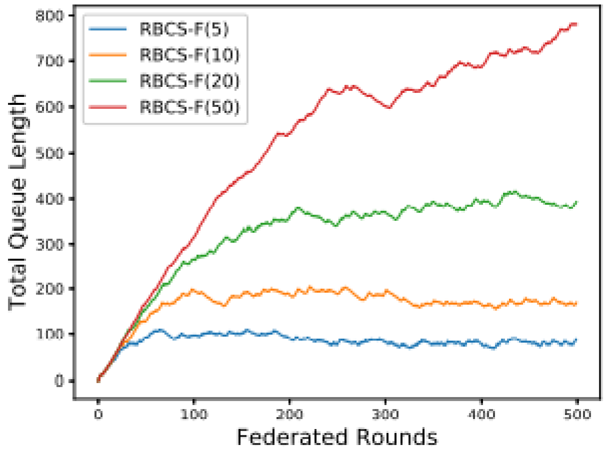


Fig. 2. The impact of V on the convergence of queues

- 不管V取什么值，The queue length最后都会收敛，也就是说V取任何值，都不会破坏公平性约束
- V值越大，The queue length收敛的速度越慢且收敛的值越大，意味着虽然长期公平性可以保证，但是前期会破坏公平性

Training time vs Client Selection strategies

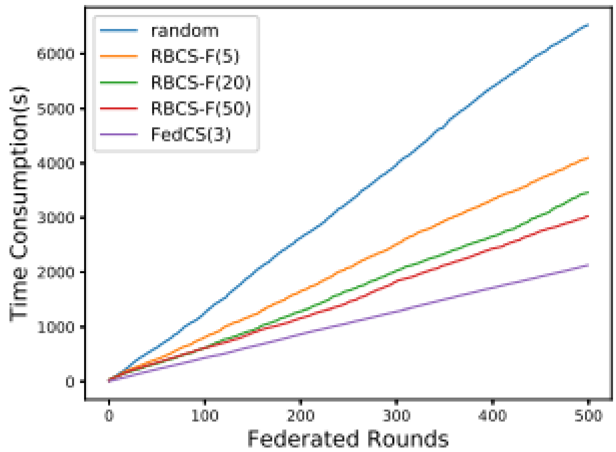


Fig. 3. Training time of different client-selection strategies

- RBCS-F的训练时间相比随机策略有较大的改进，但和FedCS之间仍有一定的Gap

> 文章指出和FedCS之间的gap是不可避免的，因为RBCS-F引入了公平因素以及在线学习的开销

- 惩罚因子V越大,训练时间越短

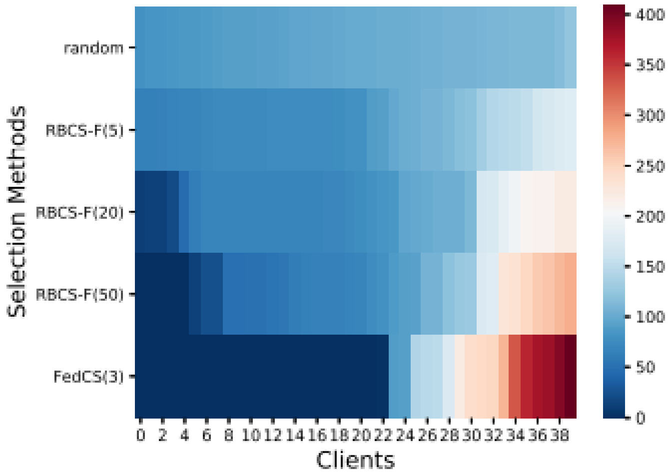
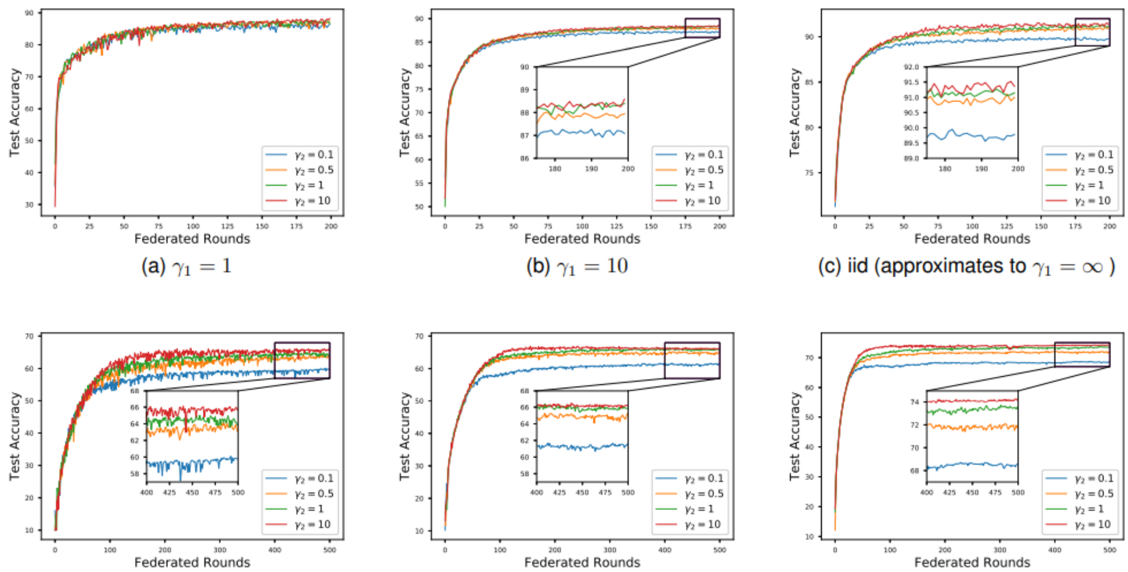


Fig. 4. Pull record of arms (or clients) under different client-selection strategies

The Model accuracy vs The Fairness factor



(a) $\gamma_1 = 1$ (b) $\gamma_1 = 10$ (c) iid (approximates to $\gamma_1 = \infty$)

(d) $\gamma_1 = 1$ (e) $\gamma_1 = 10$ (f) iid (approximates to $\gamma_1 = \infty$)

Fig. 5. Fairness impact under fashion-MNIST ((a), (b) and (c)) and CIFAR-10 ((d), (e) and (f))

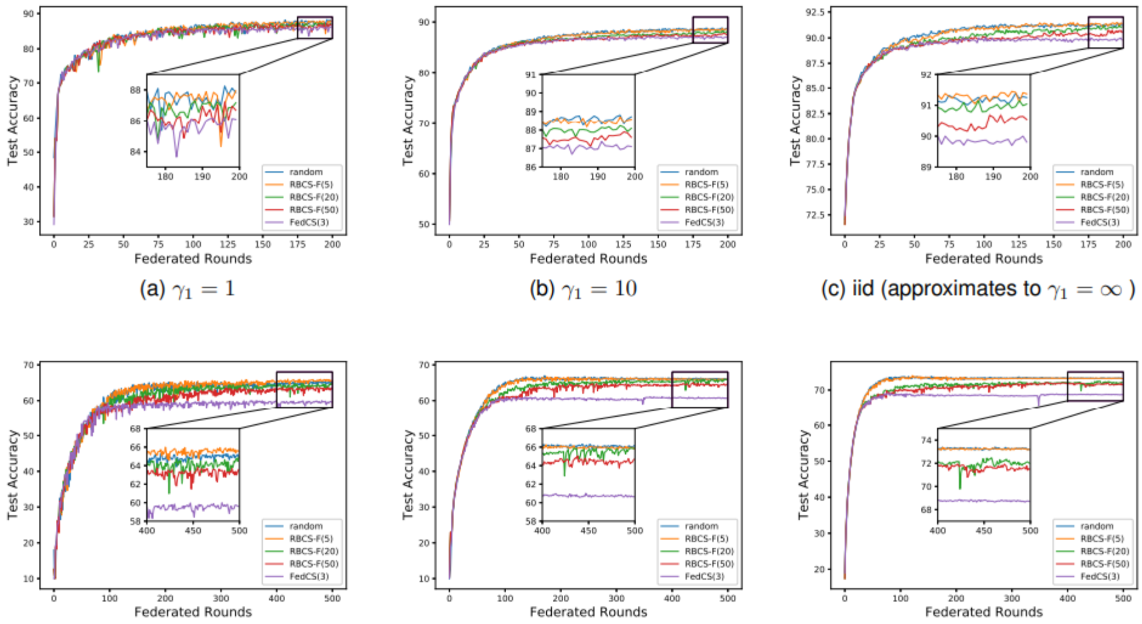
γ_1表示client之间数据的离散程度,值越小表示离散程度越大

γ_2表示fairness的离散程度，值越小表示越不公平

- 选择策略越公平，model accuracy越高

  > 在CIFAR-10的训练中更加明显，作者猜测在越复杂的任务中公平因素的影响越大，因为训练更复杂的任务可能需要更多样化的数据

- 数据non-iid的程度越大，模型的稳定性越差，且收敛需要的round越多，且non-iid的程度对公平性几乎没有影响



(a) $\gamma_1 = 1$ (b) $\gamma_1 = 10$ (c) iid (approximates to $\gamma_1 = \infty$)

(d) $\gamma_1 = 1$ (e) $\gamma_1 = 10$ (f) iid (approximates to $\gamma_1 = \infty$)

Fig. 6. Accuracy vs. federated rounds for fashion-MNIST ($(a), (b), (c)$) and CIFAR-10 ($(d), (e), (f)$)

conclusion

fairness is indeed playing a critical role in the training process.In particular, we show that a fairer strategy could promise us a higher final accuracy while inevitably sacrificing a few training efficiency.