SmartPC: Hierarchical Pace Control in Real-Time Federated Learning System

Li Li*, Haoyi Xiong*, Zhishan Guo, Jun Wang and ChengZhong Xu

ShenZhen Institute of Advanced Technology, Chinese Academy of Science
Big Data Lab (BDL), Baidu, Inc., Beijing, China
Department of Electrical and Computer Engineering, University of Central Florida
School of Computer Science, McGill University
Department of Computer and Information Science, University of Macau

*Equal Contribution
Mobile Devices Improve Fast!

- Screens are getting sharper!
- Processors are getting faster!
- Different I/O components!

Learning predictive models can effectively improve user experience!
Cloud-centric ML for Mobile

Key Problem: Data Privacy
Federated Learning

- Federated learning is proposed to collaboratively train a shared model while preserving data privacy.

**Step 1:** Initialization
**Step 2:** Broadcast
**Step 3:** Train
**Step 4:** Aggregation
**Step 5:** Converge
Critical Issues

- Federated learning can be costly on mobile devices.

- From the perspective of energy consumption:
  - Highly energy demanding and hurts the battery lifetime.
  - **Current Solution**: The training is only conducted when the smartphone is being charged.
  - Machine learning in **real-time** is becoming more and more important!

- From the perspective of training completion time:
  - Hardware **Heterogeneity** + **Unbalanced** Training Data.
  - **Synchronous** model averaging approach.
  - Overall training process is **bottle-necked** by the less powerful device.
Closely Related Work

■ Distributed Learning
  ■ Task scheduling and job placement. [INFOCOM' 19][SOCC' 17]
  ■ Distributed learning acceleration. [ICDCS' 17][OSDI' 14]
  ■ More limitation of battery lifetime and heterogeneity on mobile devices.

■ Federated Learning
  ■ Reducing the communication cost. [NIPS' 16]
  ■ Improving the security during the training process. [CCS' 17][EUROSYS' 19]
  ■ Federated multi-task learning. [NIPS' 17]
  ■ Trade-off between energy efficiency and training progress is not considered.

Balance Model Accuracy, Training Progress and Energy Efficiency in real time is highly required!
Observation 1

- A certain percentage of successful weight updates is enough to guarantee the predictive accuracy.
Observation 2

- Default DVFS governor does **not well** balance the training progress and energy in a Federated Learning system.
Observation 3

- The concurrent running foreground apps can **highly impact** the background local training process.

![Graph showing comparison of different scenarios](image)
Major Contributions of Our Paper

- **SmartPC**: a hierarchical pace control framework in real-time federated learning system.
- **First** work that balances model accuracy, training progress and energy efficiency.
- Multi-layer Design: **Global Pace Control** and **Local Pace Control**.
System Workflow
Global Pace Control

- Assigns the training deadline for each round.
- Balance model accuracy and the overall training progress.

**Initial Time Estimation:** Estimate the local training time (hardware configuration info + training data size).

**Comp Time Prediction:** Predict the training speed of the upcoming round (impact of user interaction).

**Deadline Determination:** Shortest time (a predefined percentage of participating devices can send back their weight updates).
Local Pace Control

- Meets the **deadline** while minimizing the **energy consumption** of the local training process.

**Speed Determination**: Optimal speed (in terms of IPS) that allows the deadline to be met with minimal energy.

**Resource Scheduler**: Computes a **resource schedule** for the device to balance the power and performance.

**Local Speed Monitor**: Monitors the average training speed for deadline determination of next round.
More on Local Pace Control

- Takes the IPS target as input, computes schedules for the device hardware resources to achieve the target in an energy efficient way.

- Integral Controller: Minimize the accumulated error $e(n)$ between the static target performance $r$ and the measured performance $y(n)$.

- Energy Optimizer: Determines the energy-optimal CPU frequency schedule that achieves $s(n)$. 

![](image.png)
Experiment Setup

Hardware Testbed:
- A prototype using Android smartphones with different hardware configurations.
- A monsoon power monitor is used to measure power consumption.

Simulation Testbed:
- 100 smartphones with hardware configurations randomly selected from Table I.
- Adopt usage trace from LiveLab to emulate user interaction.

### Table I: Device Profiles

<table>
<thead>
<tr>
<th>Device</th>
<th>Android Version</th>
<th># Core</th>
<th>CPU Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Honor</td>
<td>8.0</td>
<td>8</td>
<td>1.4-2.11 GHz</td>
</tr>
<tr>
<td>Lenovo</td>
<td>5.0.2</td>
<td>4</td>
<td>0.29-1.04 GHz</td>
</tr>
<tr>
<td>ZTE</td>
<td>5.1.1</td>
<td>4</td>
<td>0.20-1.09 GHz</td>
</tr>
<tr>
<td>Mi</td>
<td>5.1.1</td>
<td>6</td>
<td>0.46-1.44 GHz</td>
</tr>
<tr>
<td>Nexus 6</td>
<td>6.0</td>
<td>4</td>
<td>0.3-2.6 GHz</td>
</tr>
</tbody>
</table>
Baseline for Comparison

- Default:
  - State-of-the-practice Federated Learning system.
  - Completes the local training with default CPU governor.
  - Conduct model average when all the updates are received.

- Train-with-all:
  - Enters next round when all the updates are received.
  - Energy optimization is performed on local training.

- Fixed-deadline:
  - The deadline for each round is fixed.
  - Energy optimization is conducted on the local side.
Simulation Experiment

**Submission Rate:**
SmartPC effectively converges to the predefined value.

**Training Completion Time:**
SmartPC accelerates 2.27X on average.

**Energy Consumption:**
Achieves 28.4% energy saving on average.
Hardware Experiment

**Energy Consumption:**

**Default:** $P_{\text{train}}=3517.57\text{mW}$, $t_{\text{train}}=90.2\text{s}$, $p_{\text{idle}}=27\text{mW}$.

**SmartPC:** $P_{\text{train}}=324.7\text{mW}$, $t_{\text{train}}=689\text{s}$.

32.8% Energy Saving

**Frequency Selection:**

**Default:** 51.8% of time at level 18, 24.3% of time at level 17.

**SmartPC:** 87% at level 1.

More Experiment Results in the Paper!
Conclusion

**SmartPC**:  
- Hierarchical online pace control framework for real-time federated learning.  
- Balances the training time, model accuracy and energy efficiency.

**SmartPC** outperforms the baseline:  
- Up to 32.8% energy saving on mobile devices.  
- A speedup of 2.27 in training time.  
- Without model accuracy degradation.
Thank You !
Backup: Global Pace Control

- Completion Time Model:

\[ t_i = \frac{c_i D_i}{f_i} \]

- **c**: number of CPU cycles required to process one data object.
- **D**: number of data objects in the local training data set.

- Completion Time Prediction:

\[ r_i^j = \frac{S_i^j}{t_{i\text{-end}}^j - t_{i\text{-start}}^j} \]

\[ R_i^k = \begin{cases} r_i^1, & k = 1 \\ \alpha * r_i^{k-1} + (1 - \alpha) * R_i^{k-1}, & k > 1 \end{cases} \]

- Deadline Determination:

\[ \text{Min}\{d_k\} \]

subject to \[ \sum_{i=1}^{N} I(t_i^k) \geq U_{\text{required}} \]

\[ I(t_i^k) = \begin{cases} 0, & t_i^k > d_k \\ 1, & t_i^k \leq d_k \end{cases} \]
Backup: Local Pace Control

- Speed Determination

\[
E = p^{\text{train}} \cdot t^{\text{train}} + p^{\text{idle}} \cdot t^{\text{idle}}
\]

argmin \( E_i(f_i), \quad f_i^{\text{min}} \leq f_i \leq f_i^{\text{max}} \)

s.t. \( t_i^{\text{idle}} + t_i^{\text{train}}(f_i) = d_k, \quad 0 \leq t^{\text{idle}}, t^{\text{train}}(f_i) \leq d_k \)

- Energy Optimizer

minimize \( \sum_{i=1}^{N} \tau_i \cdot p_i \)

subject to \( \sum_{i=1}^{N} \tau_i \cdot s_i = s(n) \)

and \( \sum_{i=1}^{N} \tau_i = T, \quad 0 \leq \tau_i \leq T \)
SmartPC achieves 28.4% energy savings on average, because of intelligent pace control in the local layer.

Table II: Model Accuracy

<table>
<thead>
<tr>
<th></th>
<th>2D_CNN</th>
<th>Lenet-5</th>
<th>AlexNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>97.61%</td>
<td>97.49%</td>
<td>97.5%</td>
</tr>
<tr>
<td>SmartPC</td>
<td>97.46%</td>
<td>97.24%</td>
<td>97.35%</td>
</tr>
</tbody>
</table>
With AngryBirds: 51% of time on level 1 and 45% of the time on level 3.

With Basketball: 37% of time on level 1 and 59% of the time on level 3.
Backup: Impact of Foreground Apps

**S1** - AngryBird only
**S2** - AngryBird Training
**S3** - BasketBall only
**S4** - BasketBall Training

**Frame per Second**
- **S1**: 59.41
- **S2**: 59.23
- **S3**: 56.14
- **S4**: 52.21

**Energy Saving**
- **S2**: 17.1%
- **S4**: 14.3%