Towards Low-Energy Adaptive Personalization for Resource-Constrained Devices

EuroMLSys’ 2024

Yushan Huang, Josh Millar, Yuxuan Long, Yuchen Zhao, Hamed Haddadi

22/04/2024
Contents

1. Motivation
2. Experimental Setup
3. Results
4. Limitations + Future Work
5. Summary
Motivation

- On-device ML is vulnerable to distribution shift.
- Can group distribution shifts into 3 types: input-, feature-, and output-level [1].
- Currently, most fine-tuning (FT) approaches involve FT a full model or its last layer(s).

Fig 1. Examples of the 3 types of distribution shift.
Target Block Fine-Tuning (TBFT):

- fine-tunes model “blocks” corresponding to the type of shift present
- e.g., last block(s) for output-level shift.
- achieves performance accuracy than full/last-layer FT at reduced/equivalent energy cost.
Experimental Setup

- **ResNet-26**: inherent “block” structure.
- TBFT on 3 datasets, one for each shift (i.e., input-, feature-, and output-level).
- Set training-sizes to 10%, 20%, and 30% of full dataset (validation- and test-sizes are both 10%).

![Diagram of ResNet-26](image)

**Fig 2.** ResNet-26 and its “block” structure.
Results: Input-Level

- Source: CIFAR10
  - 32x32 images with 10 classes.
- Target: CIFAR10-C
  - corrupted images from same classes
  - 14 corruption types (e.g., Gaussian noise, motion blur, brightness).
- Performance averaged across corruption types.

Fig 3. Example corruptions from CIFAR10-C.
Fig 4. Results of TBFT on CIFAR10-C.

The horizontal dashed line shows the accuracy of full FT.
Results: Feature-Level

Living17:
- ImageNet-based i.e., 128x128 images
- Created as part of BREEDS [2]
- Living things across 17 classes, each with 4 different subclasses
- Source and Target distributions consist of different subclasses.

![Example Source and Target images](Fig 5)
Results: Feature-Level

Fig 6. Results of TBFT on Living17.
Results: Output-Level

- Source: CIFAR10
  - 32x32 images with 10 classes.
- Target: CIFAR10-flip
  - images from same classes with labels flipped.
  - i.e., label $y$ becomes $9-y$

Fig 7. Results of TBFT on CIFAR10-flip.
Results: Energy Cost

- Device: Raspberry Pi 4 Model B
- Metrics:
  - Energy Cost ($E$), derived from $E = Pt$ (where $P$ is power and $t$ is runtime)
  - $P$ ($W$) recorded using Power Monitoring HAT [3]
  - Energy Saving ($ES$) vs full FT, calculated as $ES = (E_{TBFT} - E_{FT})/E_{FT}$

<table>
<thead>
<tr>
<th></th>
<th>Resolution 32, Output 10</th>
<th>Resolution 128, Output 17</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Block 1</td>
<td>Block 2</td>
</tr>
<tr>
<td>Time (s)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.45</td>
<td>0.33</td>
</tr>
<tr>
<td>Energy Cost (J)</td>
<td>0.090</td>
<td>0.066</td>
</tr>
<tr>
<td>Energy Saving (%)</td>
<td>15.09</td>
<td>37.74</td>
</tr>
</tbody>
</table>

Compared to full model FT, TBFT results in an average energy-saving of 41.57%.
Limitations + Future Work

• **Shift Type Discrimination**
  • TBFT relies on knowledge of shift type: unrealistic.
  • Parameter selection methods are gradient-based and expensive.
    • instead, select based on incoming data

• **Unsupervised Personalization Methods**: essential for realistic on-device learning.

• **Multidimensional Personalization**
  • complex and composite shifts in real applications (e.g., coinciding input- and feature-level shifts).
We categorize distribution shift into 3 types: input-, feature-, and output-level.
TBFT fine-tunes model blocks corresponding to distribution shift type.
TBFT achieves better performance than full/last-layer FT with reduced/equivalent energy-consumption.

- avg. **15.3% improvement** in **accuracy** over full FT.
- avg. **41.6% improvement** in **energy-consumption** over full FT.

Thank you