On-Device Training with Local Sparsity for Federated Learning

Xinchu Qi1, Javier Fernandez-Marques2, Pedro PB Gusmao1, Yan Gao1, Titouan Parcollet3 and Nicholas D. Lane1
1University of Cambridge, 2University of Oxford, 3Université Avignon

TL;DR
- We explore training with highly sparse tensors in FL clients to accelerate compute. We achieve this by replacing all dense convolutions with sparse-dense convolutions. These can be accelerated for a sufficiently high sparsity ratio.
- While sparse training has attracted some attention, it has not been investigated in the context of Federated Learning.
- While the resulting models are not sparse, we identify that the locations of high magnitude weights remains constant. We introduce a masking mechanism to exploit this observation and save on up-link communication.

Federated Learning: Background and Challenges

- FL is a form of distributed ML where nodes are commodity devices such as smartphones, wearables or other IoT devices.
- The sophistication of the models that FL clients can train is constrained by the compute and memory limitations of the clients and the associated communications costs.
- These challenges has been partially eased by mechanisms relying on pruning, quantisation and distillation. These three techniques have dominated the recent "efficient FL" literature.
- Our method replaces all dense convolutions with sparse-dense convolutions in both forward and backward passes.
- We adapt SWAT (Raihan & Aamodt, 2020) to the FL setting and note that naively introducing sparsity result in severe degradation compared to centralised training.

Algorithm 1 ZeroFL. Let us consider a cluster of $N$ total clients on local data set and each with a learning rate $\eta$ and round $T$ with $T$ the total number of communication rounds. The client has the data set $u_i$. The number of clients participating in each round is denoted as $K$. $w_i$ represent all the weights aggregated at round $t$ and $d_t$ the difference of weights.

Forward Pass
- Layer Input
- Layer Weights
- Sparse Input Tensor
- Forward
- Layer Output
- Sparse
- TPR

Central server does:
for $t = 0, \ldots, T − 1$ do
Server randomly selects $K$ devices.
for all $k$ in $K$ do
Perform TrainLocally($t$, $w_k$);
Aggregation:
if Top-K-Weight then $w_{t+1} ← w_{t+1} + \sum_{k=1}^{K} d_{t,k}$
if Diff on Top-K-Weight then $w_{t+1} ← w_{t+1} + \sum_{k=1}^{K} d_{t,k}^2$
if Top-K Diff then $w_{t+1} ← w_{t} + \sum_{k=1}^{K} d_{t,k}^2$
TrainLocally($t$, $w_k$);
for $i = 1, \ldots, E$ do
Do local model training via SWAT with sparsity level $sp$.
$w_i ← w_{t+1} − \eta_i \nabla F(w_{t+1})$
Determine which weights to send for aggregation:
if Top-K-Weight then return top $1 − sp + \epsilon_{\text{mask}}$ Weights.
$d_{t+1}^k ← \epsilon_{\text{mask}} − w_i$
if Diff on Top-K-Weight then return $d_{t+1}^k$ of top $1 − sp + \epsilon_{\text{mask}}$ weights.
if Top-K-Weights Diff then return top $1 − sp + \epsilon_{\text{mask}}$ of $d_{t+1}^k$.

Results
- Study the impact of ZeroFL sparse training and masking in both IID and non-IID settings for image classification (CIFAR-10, FEMNIST) and keyword spotting audio classification (Speech Commands).
- Masking balances communication and training/aggregation quality.
- We observe that high-magnitude weights remain at fixed location throughout training. By masking small values, the performance of the global model increases and communication savings are obtained.

<table>
<thead>
<tr>
<th>non-IID</th>
<th>Sparsity Level</th>
<th>SWAT Full Model</th>
<th>ZeroFL (m=0.2)</th>
<th>File Size (MB)</th>
<th>Comms Save</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>90%</td>
<td>80.62%</td>
<td>81.04%</td>
<td>27.3</td>
<td>1.6x</td>
</tr>
<tr>
<td>Speech Commands</td>
<td>90%</td>
<td>74.00%</td>
<td>75.54%</td>
<td>23.0</td>
<td>1.9x</td>
</tr>
<tr>
<td>FEMNIST</td>
<td>90%</td>
<td>82.81%</td>
<td>84.90%</td>
<td>27.3</td>
<td>1.6x</td>
</tr>
</tbody>
</table>

Comms Save