# Exploring Approximate Computation Reuse in Federated Learning

Yunxiao Du#, Zichen Xu#, Xun Jiao*

#Nanchang University *Villanova University

## Introduction

Artificial Neural Network characterizes complex data inputs by repeating propagation on the network model. Federated learning extends such ANN training to multiple participants for efficiency and data privacy. However, processing federated learning can be expensive due to the intensive workloads and limited resources ( e.g., energy ) available in the subset of these participants. Approximate computation usually reuses some arithmetic instructions in a hybrid precision manner thus can cheapen federated learning. In this paper, we explore the possibility of enabling approximate computation reuse in federated learning and harvesting potential energy savings.

We share our observations on several entries of applying Bloom filters with arithmetic units during the learning process. BFs store and recall frequently occurring input patterns to reuse computations. Therefore we can harvest the opportunities for computation reuse by storing frequent input patterns specific to a given layer and using approximate pattern matching with hashing for limited data precision in a controlled manner. We empirically studied a few strategies on approximate computation reuse for federated learning. Results show that our work can finish the learning process successfully, consuming 47% less energy on average, pays negligible overhead (average 0.1%), and accuracy loss (3% at most), as compared to the default setting.

## Federated Learning & Bloom Filter

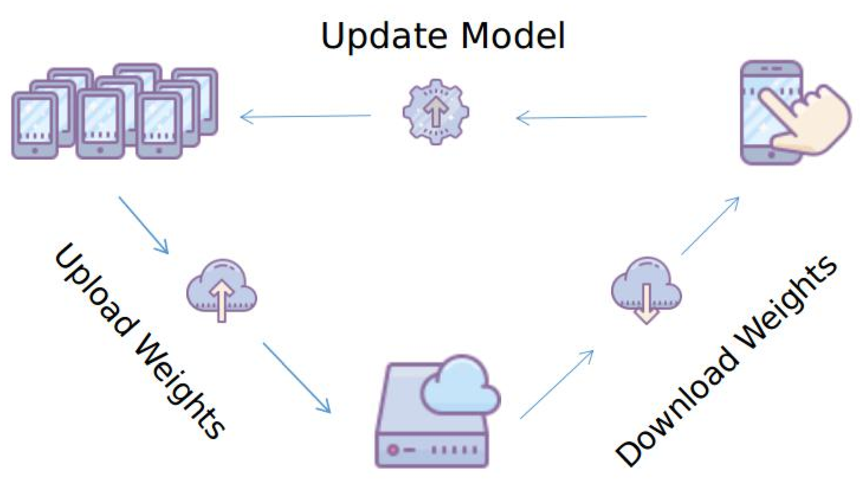


Fig. 1 The State Transition Diagram of Federated Learning.

**Step 1 :** With a specific learning Job, the leader node heartbeats any connected candidates and initializes them as workers.
**Step 2 :** After the initialization, the leader node broadcasts the current global model to all workers, and starts the learning process.
**Step 3 :** All workers train their local data using the current model for several iterations.
**Step 4 :** At the end of each epoch, the leader aggregates model gradients and applies the update. Based on the weighted average, the leader sends the updated model difference to all workers.
**Step 5 :** Repeat Step 2-4 until the global model converges.

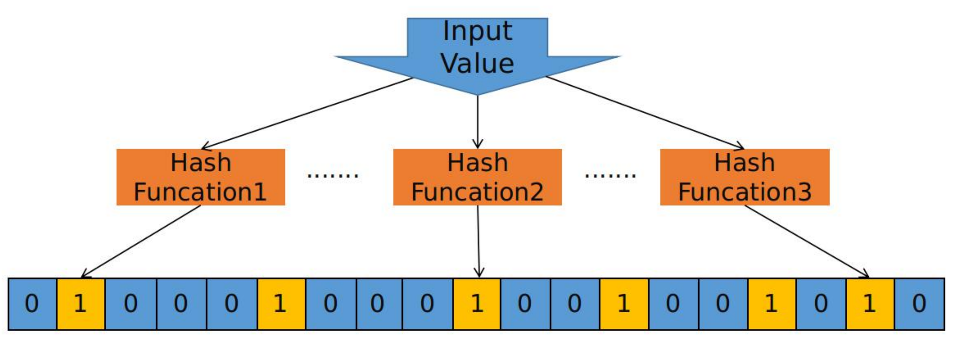


Fig. 2 A Simple Lyout of Bloom Filter.

BF is efficient since it removes elements out of the target set, however, it surfers from false positive, which in our case, hurts the overall modeling accuracy. Bloom can be directly integrated directly into the hardware. In our current setup, we evaluate it using compiler techniques and place the hardware implementation as our future work.

## Impact of Design Factors

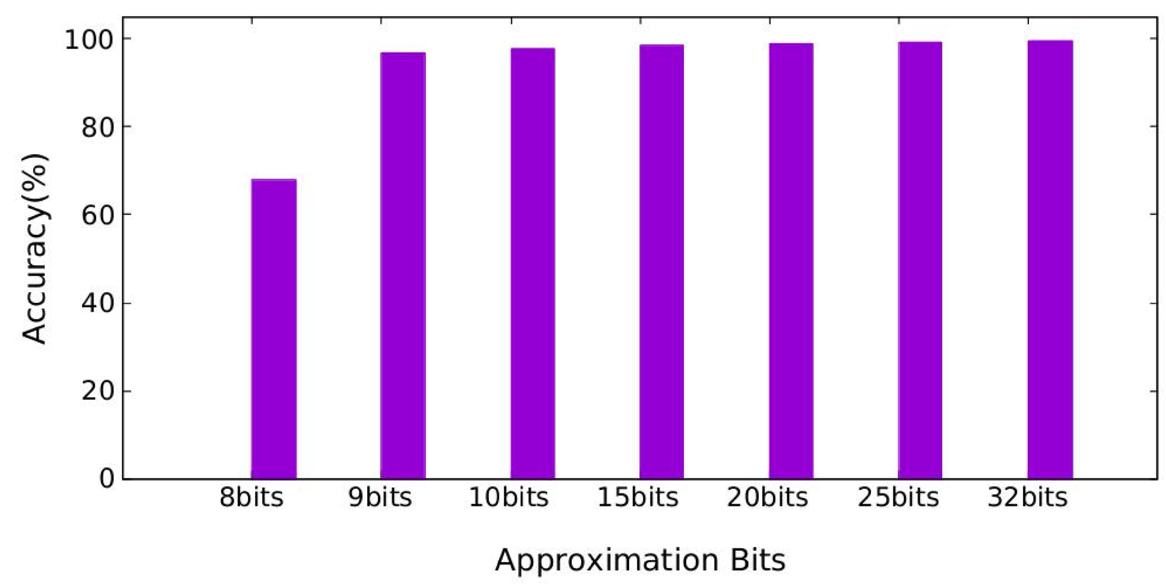


Fig. 3 Accuracy loss under different bit approximation matching.

## Energy savings

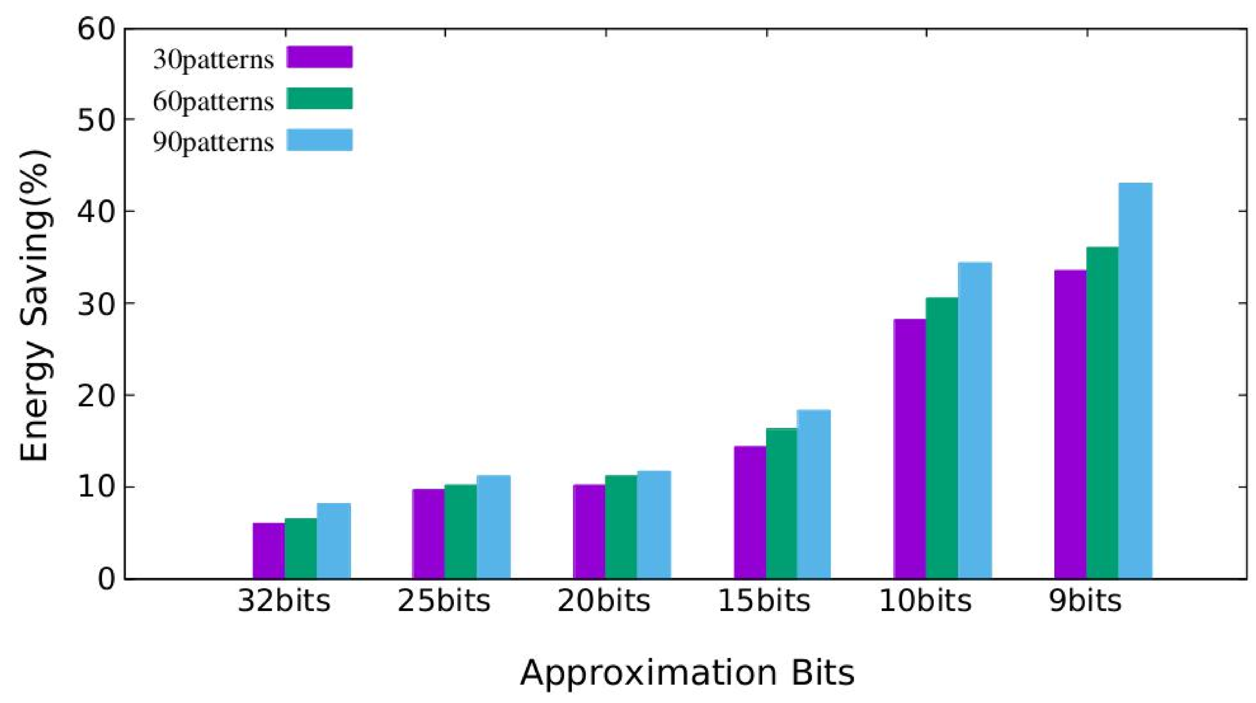


Fig. 4 Energy Savings in Different Situations

## Take Home Message

In practice, we have prototyped and evaluated our design onto a framework for Android devices and a test board with a translation lookaside buffer for implementing Bloom filter. Our results show that our work can finish the learning process successfully with limited patterns, consuming 47% less energy on average, and pays negligible overhead (average 0.1%) and accuracy loss (3% at most), as compared tostate of the arts. As such, we prove it is possible to retain federated learning for smartphones. In this paper :

- ◆ We identified the performance problem of federated learning on devices.
- ◆ We conducted a series of experiments on understanding the computation behavior of federated learning.
- ◆ We propose a mechanism based on our observations, making efficient on-device federated learning.
- ◆ We prototyped our work on physical testbed and verified our solution.

### Future Works:

- ● Implementing the hardware layout combined with modern smartphones and performing more empirical studies on more models.
- ● Exploring the possibility of approximate calculations on GPU.

### Reference:

[1] Burton H Bloom. Space/time trade-offs in hash coding with allowable errors. Communications of the ACM, 13(7):422–426, 1970.
[2] Vahideh Akhlaghi, Abbas Rahimi, and Rajesh K Gupta. Resistive bloom filters: from approximate membership to approximate computing with bounded errors. In 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), pages 1441–1444. IEEE, 2016.
[3] Swagath Venkataramani, Ashish Ranjan, Kaushik Roy, and Anand Raghunathan. Axnn:energy-efficient neuromorphic systems using approximate computing. In Proceedings of the 2014 international symposium on Low power electronics and design, pages 27–32. ACM, 2014.