Accelerating XGBoost on ARM CPUs with Scalable Vector Extension for High-Performance Data Science

Divya Kotadiya, Ragesh Hajela, Doteguchi Masahiro, Priyanka Sharma
Fujitsu Limited

Abstract

- Decision trees are a cornerstone of many machine learning algorithms, offering interpretable & robust models for structured data. **XGBoost (eXtreme Gradient Boosting)**[1] uses an ensemble of decision trees to deliver high performance in gradient boosting.
- In this work, we leverage ARM **Scalable Vector Extension (SVE)**[2], which is a vector extension for Armv-8A that supports variable length vectors from 128 to 2048 bits. By utilizing SVE's vectorization capabilities we accelerate XGBoost's training pipeline by optimizing the histogram update function - a key step in constructing decision trees.
- The results of our experiments on **Higgs Boson dataset** show a **2x speed-up** in training time compared to the non-SVE optimized code, with same accuracy on **ARM** architectures.

Methodology

Figure 1 illustrates implementation of gradient boosting, where for a given dataset X, subsequent trees are built by learning the residual of the previous tree as the input for the next tree.

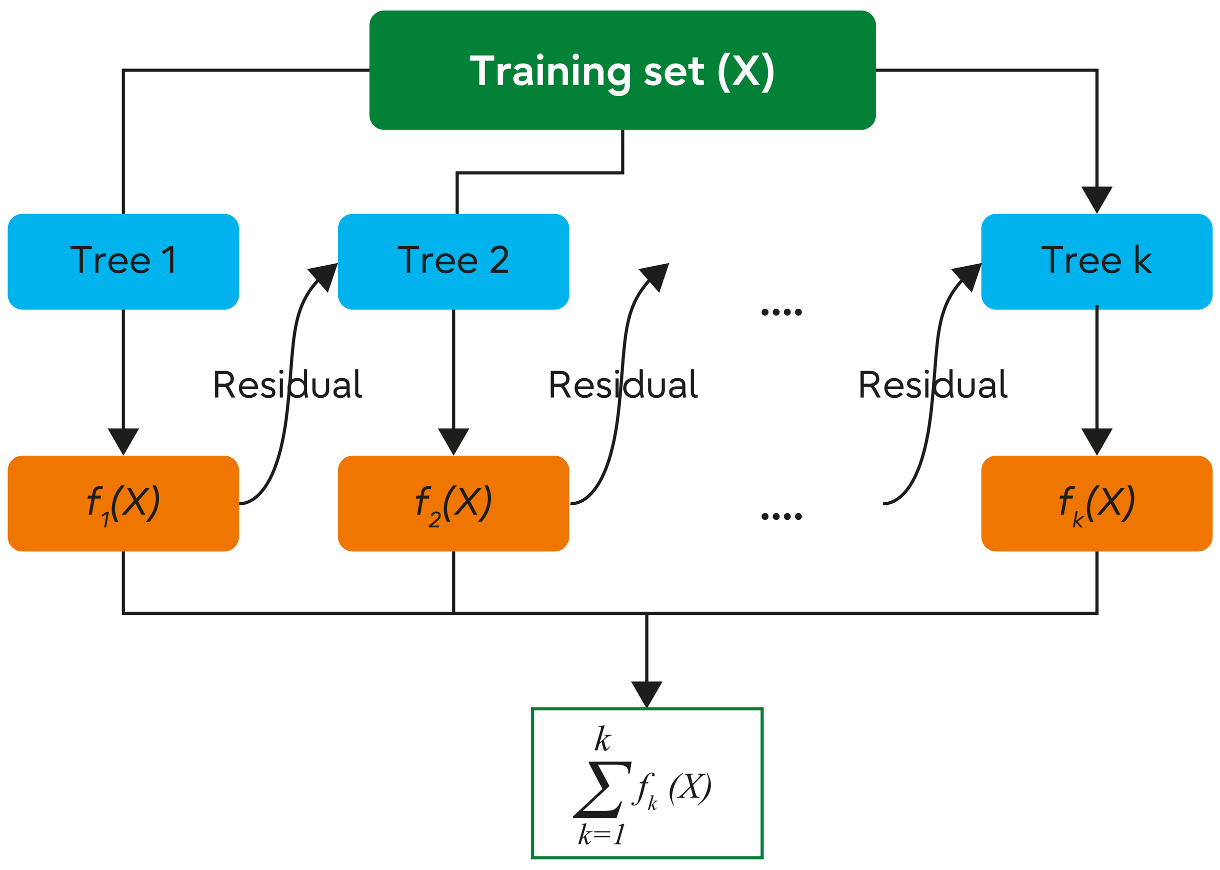

Figure 1 : Implementation of gradient boosting in XGBoost

Figure 2 outlines the key steps in training pipeline, highlighting the function which is optimised using SVE to compute and update histogram values efficiently.

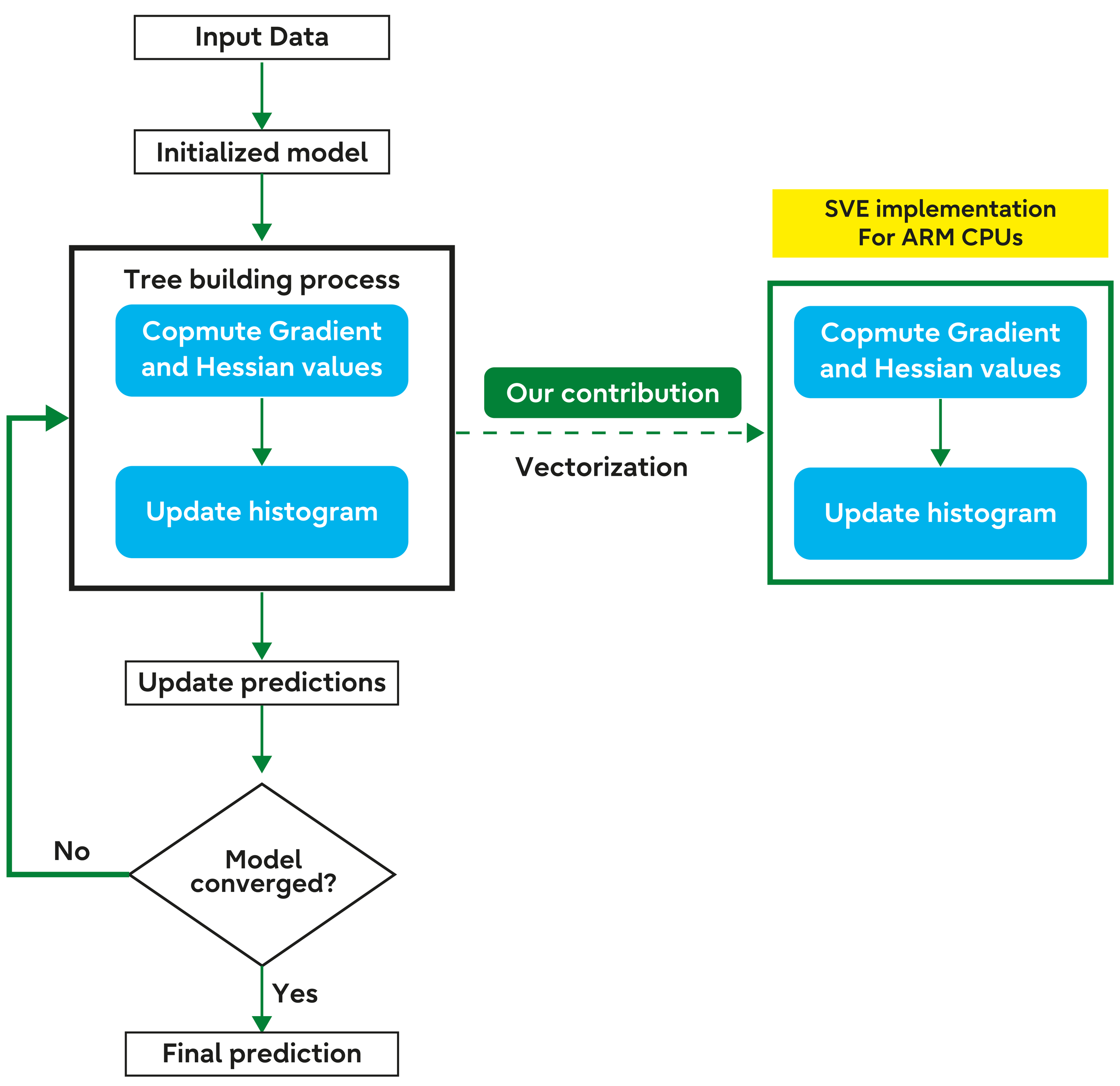

Figure 2 : Key steps in XGBoost training pipeline

The following features of SVE contributed in boosting the performance of the XGBoost training process on ARM :

1. Predicate-Based Operations: Predicates control active lanes by creating masks to process only valid elements ensuring safe and efficient computation at the boundaries of the dataset.

2. Gather-load & Scatter-store operations: Parallel access to non-contiguous memory reduces memory bottlenecks, which is crucial for decision tree building process.

3. Parallelism for Scalability: SVE maximizes parallelism by processing multiple data elements simultaneously, reducing loop iterations and computation time.

Figure 3 showcases the pseudocode of the function modified using SVE intrinsics.

```
Pseudocode for UpdateHistogramWithSVE

Function: UpdateHistogramWithSVE
Steps:

1  Load gradient and Hessian values from the input array
2  For each chunk of rows with size equal to vector width:
   (a)  Create masks for valid elements in the current chunk
   (b)  Load gradient index values and offsets for the current chunk
   (c)  Calculate index

     • If there are any missing values, multiply gradient index by 2
     • Otherwise, add the offsets to the gradient index and then
       multiply by 2

   (d)  Split 32-bit index vector into two 64-bit vectors (lower and
        upper)
   (e)  Increment indices for Hessian values (lower + 1, upper + 1)
   (f)  Gather histogram values for computed indices
   (g)  Update Histogram: Add the gradient and hessian values to the
        histogram data
   (h)  Store updated histogram back into memory

3  End For

End Function
```

Figure 3: Pseudocode for UpdateHistogramWithSVE

Results

Figure 4 presents the performance numbers on ARM, showing **2x** speed-up with SVE implementation across multiple cores compared to the default implementation. These measurements were taken on an AWS Graviton 3 (c7g.8xl) machine with 32 cores.

Dataset details: Kaggle's Higgs Boson dataset[3]

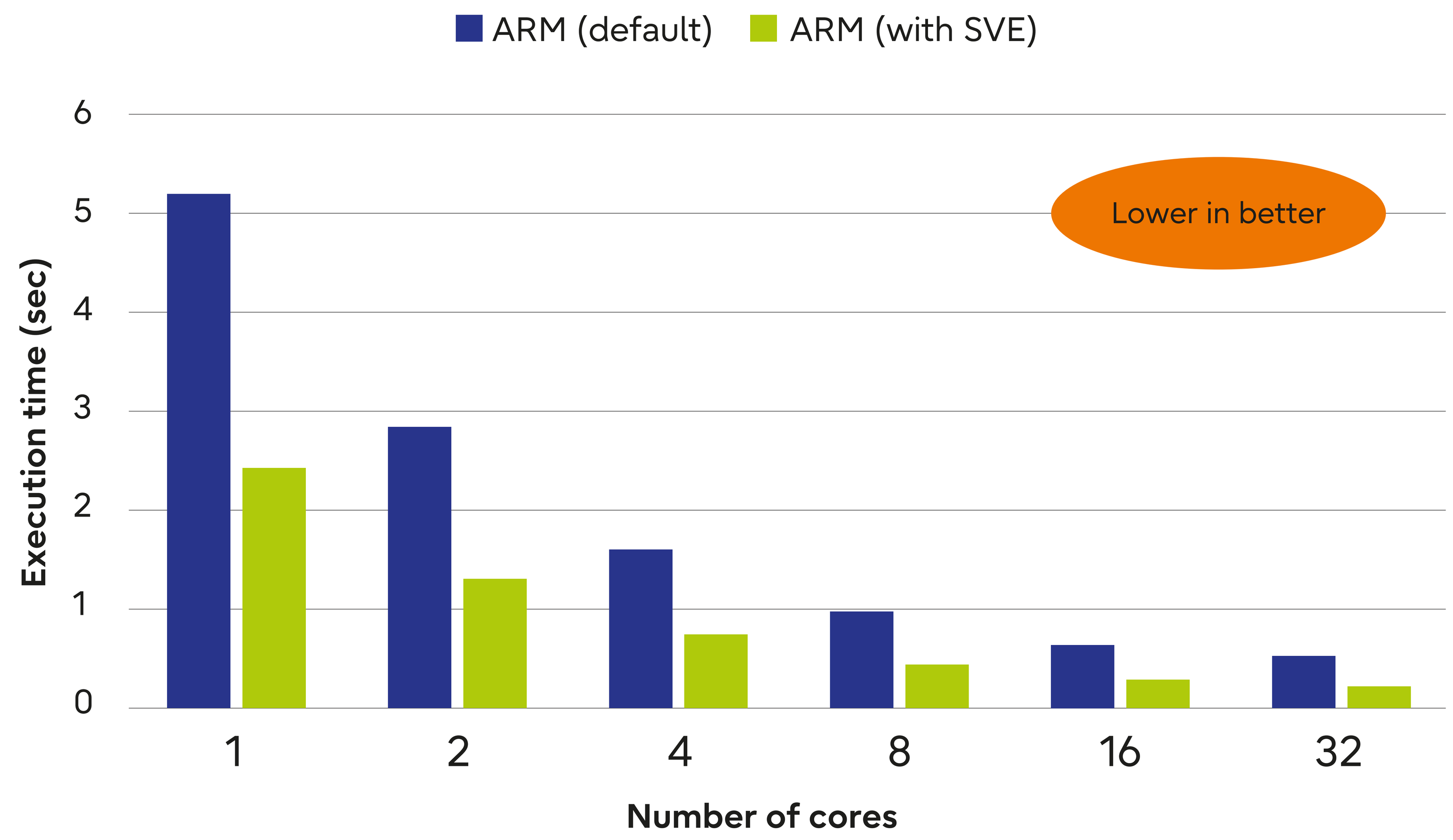

Figure 4 : Performance improvement with SVE

Conclusion and Future Work

- Our work improves the performance of training algorithm of XGBoost by leveraging SVE intrinsics, achieving up to a **2x** speed-up. The enhancements maximize hardware utilization and scalability, paving the way for high-performance gradient boosting on ARM platforms.
- This work is planned to be extended towards accelerating other gradient boosting algorithms like LightGBM, CatBoost, etc.

References

[1] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.

[2] A64 SIMD Instruction List: SVE Instructions

[3] Higgs Boson Dataset

Link to Pull request

This poster is based on results obtained from a project, JPNP21029 subsidized by the New energy and Industrial Technology Development Organization (NEDO).