Improving multithreaded performance of tree-based machine learning models

In the rapidly evolving field of machine learning (ML), the effectiveness of tools and libraries plays a crucial role in the development and deployment of models. TL2cgen, a prominent tool in this landscape, stands out for its ability to convert tree-based machine learning models into C code, facilitating efficient model deployment in various environments.

To speed up the predictions, the current version of TL2cgen uses OpenMP for parallelizing the prediction making process. However, users report a slower performance when comparing to the old Treelite runtime, which uses custom thread pools, even though the generated c code is the same. The aim of this project is to find out if OpenMP is the cause of this performance deficit, by making a custom optimal thread pool implementation in TL2cgen and comparing the performance to the original OpenMP version.

Through this research, we aim to provide the users of TL2cgen with the optimal parallel implementation of the predictor function, ensuring the fastest execution times.

Methodology:

1- Familiarize yourself with the TL2cgen library
2- Make a custom thread pool implementation of the ParallelFor function.
3- Run benchmarks on a wide variety of input data.
4- Report the performance of both implementations on the selected input data.

Theory: 30%
Coding: 20%
Evaluation: 30%
Writing: 20%

Key References:

- Tl2cgen (https://tl2cgen.readthedocs.io/en/latest/index.html)
- https://dl.acm.org/doi/pdf/10.1145/3508019
- https://link.springer.com/chapter/10.1007/978-3-031-26419-1_32
- https://github.com/dmlc/tl2cgen/issues/18 (reported performance difference)

Contact Information

- Supervisor (CAES group)
  Duncan Bart d.bart@utwente.nl, Kuan-Hsun Chen k.h.chen@utwente.nl

- Coordinator (CAES group)
  Ghayoor Gillani, s.ghayoor.gillani@utwente.nl, room ZI 5039