

Hls4ml Synthesis Testing

Caiden Swanson, William-Rainey Harper college – CCI Intern

FERMILAB-POSTER-24-0174-STUDENT

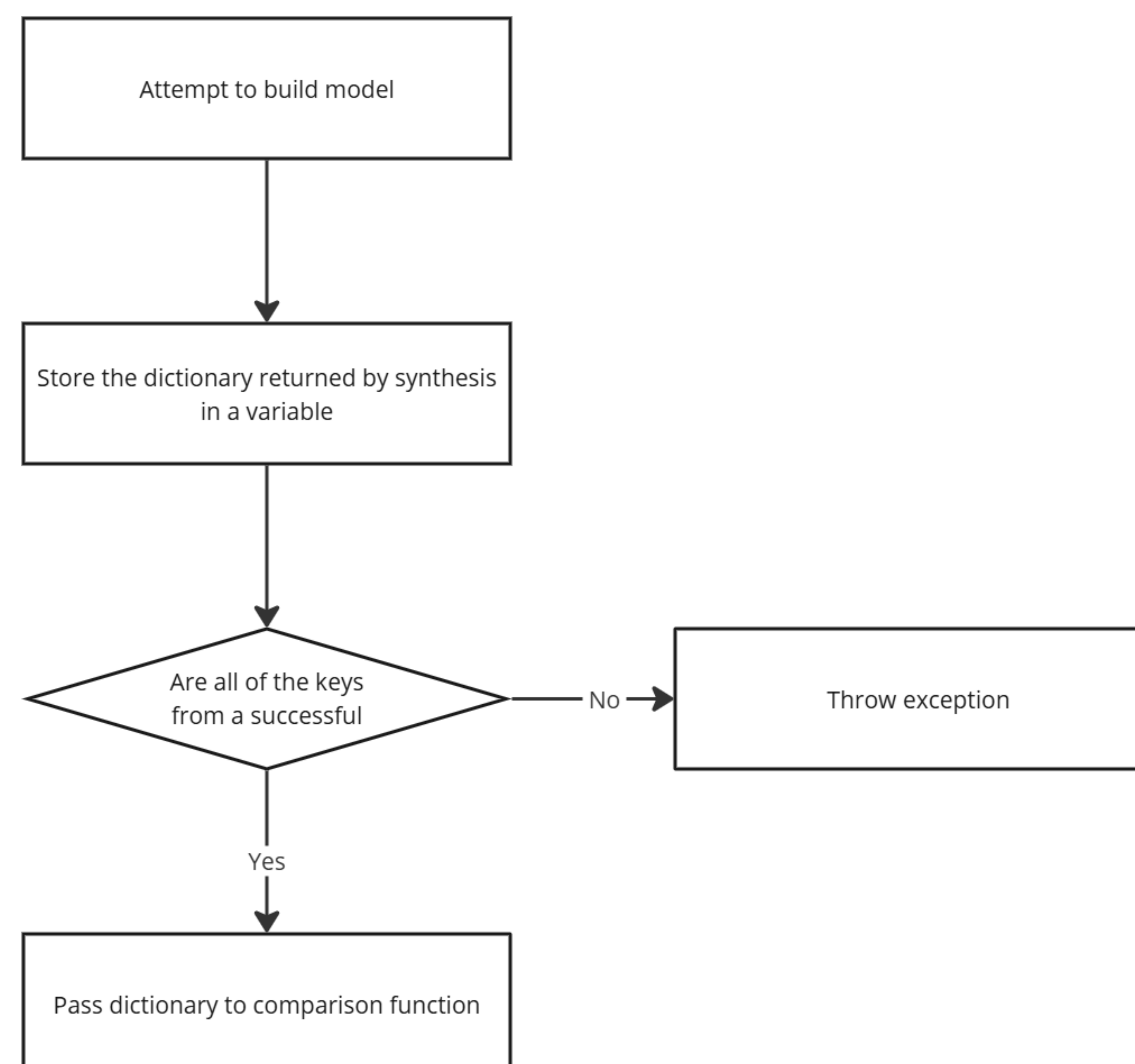
Introduction

HLS4ml (high level synthesis for machine learning) is a Python package used to translate commonly used open-source machine learning models into HLS. This is useful in machine learning applications on FPGAs. Machine learning algorithms are only as fast as the hardware that they are used on, and some applications require high speed without sacrificing accuracy. In these situations, an FPGA is a good choice since it is faster than a CPU or a GPU, but programming an FPGA is difficult. This is where HLS4ml can be used to simplify the process, as a well-known learning model can be converted to HLS and more easily deployed onto an FPGA. There are many use cases for a machine learning algorithm running on an FPGA. For example, detectors in a particle accelerator cannot keep every event that they detect, and so a computer must decide which events to keep and which to discard. Using an FPGA with a machine learning algorithm would be a good way to keep as many events as possible.



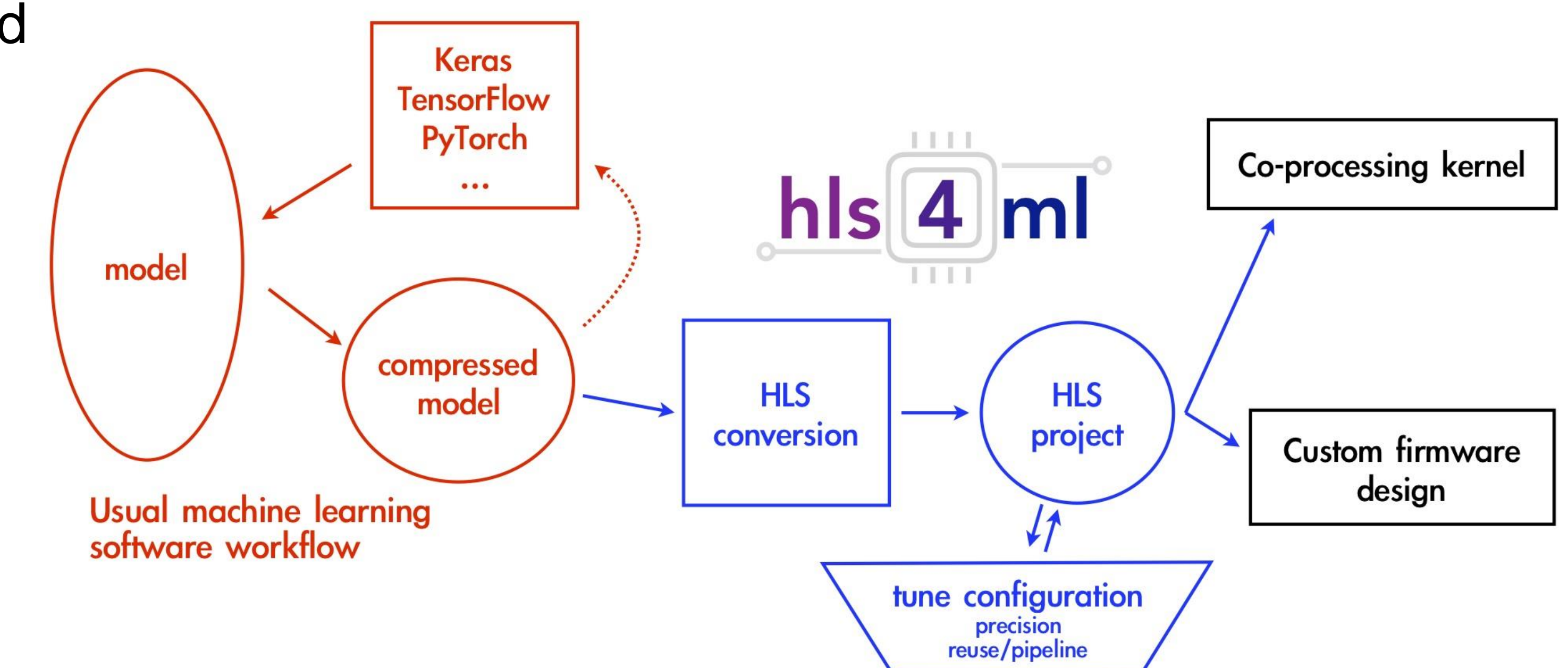
Testing

The HLS4ml package has many tests to ensure that HLS4ml is working properly. Some of them are bash scripts, and currently the synthesis tests use the bash scripts. The interface for the synthesis tests is now deprecated and doesn't have much flexibility for different backends. The majority of the tests are Python scripts, and are run using Pytest, which is used to make light and easy to read tests. So far, I have added synthesis to a couple of the Python tests and added a simple function to compare the synthesis results to a baseline. The script will soon be run from the Jenkins file. Jenkins is an automation software and is used in HLS4ml to build and test the model.



Next steps

The final goal is to allow the code maintainers to specify a directory where the baseline results are stored, and have Jenkins pass that directory to a script which compares the new results to the baseline. Currently the test that I have been working on contains synthesis as well as a way of confirming that the synthesis was successful. It also has a simple function that compares the results of the synthesis to another set of results. Eventually the comparison function will be made into its own script which will be run directly from the Jenkins file rather than being a part of a Pytest.



References

Fast Machine Learning Lab. (2024). *Hls4ml*. Retrieved from Hls4ml: <https://fastmachinelearning.org/hls4ml/index.html>

Holger Krekel and Pytest-Dev Team. (n.d.). *pytest: helps you write better programs*. Retrieved from Pytest: <https://docs.pytest.org/en/stable/>

Intel High Level Synthesis Compiler. (n.d.). Retrieved from Intel: <https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/hls-compiler.html>

Vitis HLS. (n.d.). Retrieved from AMD: <https://www.xilinx.com/products/design-tools/vitis/vitis-hls.html>

•This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.
•This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Community College Internship (CCI)