



# **Benchmark of Deep Learning models**

CERN Summer Student Programme Project Report

**Author**

Afan Terko

*afan.terko@cern.ch*

**Supervisor**

Lorenzo Moneta

*lorenzo.moneta@cern.ch*

August 2021

## Contents

|                                      |   |
|--------------------------------------|---|
| Abstract.....                        | 1 |
| Introduction .....                   | 2 |
| ROOT, TMVA, Keras.....               | 2 |
| ROOTBench .....                      | 2 |
| Benchmarking TMVA against Keras..... | 4 |
| CPU benchmarks.....                  | 4 |
| GPU benchmarks .....                 | 4 |
| Conclusion and Future work.....      | 5 |
| Acknowledgements .....               | 5 |

## Abstract

In the field of high-energy physics, software performance is critical due to the high CPU and I/O costs of processing and analyzing billions of events. Running benchmarks is therefore essential as it provides consistent performance metrics that can be tracked over time. Aim of this project was to develop benchmark code to evaluate the CPU and GPU performance for inference of some typical deep learning models used by the LHC experiments. Benchmark tests for Convolutional neural network (CNN) and Recurrent neural network (RNN) for both TMVA and Keras were created and are to be included in ROOTBench. Keras outperforms TMVA in terms of CPU performance for inference in both CNN and RNN. In terms of GPU performance, TMVA surpasses Keras in both CNN and RNN benchmarks.

## Introduction

The purpose of this project report is to summarize my work at CERN as a Summer Student. My project's goal was to create benchmark code to evaluate the CPU performance for inference of some typical deep learning models used by the LHC experiments. This benchmark code is supposed to compare the performance of different machine learning tools. The ROOT benchmark test suite (ROOTBench) will incorporate these benchmark tests. In ROOT, we have TMVA that we can use for deep learning. We would like to see how TMVA stacks up against Keras and PyTorch.

### ROOT, TMVA, Keras

ROOT is an open-source data analysis framework used by high-energy physics and others. It is developed at CERN and used by most of the experiments to store and process the data. It is written in C++ and has a C++ interpreter, making it ideal for fast prototyping. Thanks to its unique dynamic and powerful Python and C++ binding, it also integrates with Python.

The Toolkit for Multivariate Data Analysis with ROOT (TMVA) is a ROOT package that provides a machine learning environment for the processing and evaluation of multivariate classification, both binary and multi-class, and regression techniques targeting applications in high-energy physics.

Keras is an open-source software library for artificial neural networks that provides a Python interface. It serves as an interface for the TensorFlow library.

### ROOTBench

Due to the high CPU and I/O costs of processing and analyzing billions of events, software performance is critical in the field of high-energy physics.

ROOTBench is based on Google's micro benchmarking infrastructure, [gbenchmark](#). It consists of a collection of small programs that are built on top of ROOT. Their primary goal is to provide consistent performance metrics that can be tracked over time. ROOTBench uses Grafana for visualization and InfluxDB as the time series database due to Grafana's native support.

New benchmarks can be added using the following template as seen in [rootbench GitHub repository](#)

```
#include "ROOT_HEADER_TO_BENCHMARK.h"
#include "benchmark/benchmark.h"

// Replace the CLASSNAME and ROUTINE with the ROOT class and routine you
// are benchmarking respectively.
static void BM_CLASSNAME_ROUTINE(benchmark::State &state) {
  // Initialization section before actual benchmarking.
  for (auto _ : state) {
    // The benchmarking code goes here.
  }
  // Teardown.
}
BENCHMARK(BM_CLASSNAME_ROUTINE);

// In the end of the file we add our main().
BENCHMARK_MAIN();
```

The new benchmark must be registered in the system after it is added. This can be accomplished by adding an entry to the appropriate CMakeLists.txt file next to the benchmark's source code.

```
RB_ADD_GBENCHMARK(CLASSNAMEBenchmarks
  CLASSNAMEBenchmarks.cxx
  LABEL short
  LIBRARIES LIST OF LIB DEPENDENCIES)
```

## **Benchmarking TMVA against Keras**

In order to conduct representative benchmarks and make a meaningful comparison, it is important to establish a common ground between the two frameworks being compared. Thankfully, TMVA has an integrated interface that allows us to run Keras within TMVA. As a result, the data sampling and preparation processes are identical in both frameworks, TMVA and Keras.

As there are so many different deep learning models, after consulting with my supervisor, I decided to focus on CNN and RNN benchmarks. As a result, I developed CPU and GPU benchmark tests for these two deep learning models.

A convolutional neural network (CNN) is a type of artificial neural network used in deep learning to analyze visual imagery. CNN can take an input image, assign importance (learnable weights and biases) to different aspects/objects in the image, and distinguish between them. A recurrent neural network (RNN) is a type of artificial neural network that works with time series or sequential data.

### **CPU benchmarks**

To run CPU benchmarks, I used an i5-8265U CPU utilizing all eight threads. The Keras benchmark was around 80% faster than the TMVA benchmark for CNN. In terms of RNN, the Keras benchmark was faster by a factor of 3.5 than the TMVA benchmark. This suggests that Keras outperforms TMVA in terms of CPU performance for inference in both CNN and RNN, and that TMVA still has room for improvement.

### **GPU benchmarks**

To run GPU benchmarks I used Nvidia GeForce RTX 2070 GPU. The TMVA benchmark was about 30% faster than the Keras benchmark for CNN. In terms of RNN, the TMVA benchmark was faster by a factor of 3 faster than the Keras benchmark. As we can see, TMVA beats Keras in terms of GPU performance for inference in both CNN and RNN.

## **Conclusion and Future work**

We can conclude that TMVA outperforms Keras on the GPU in both CNN and RNN, whereas Keras is better suited for the CPU than TMVA. In future, it would be very useful to write PyTorch benchmarks as well. This would enable us to compare the performance of three machine learning tools: TMVA, Keras, and PyTorch. In addition, more benchmarks for deep learning models other than CNN and RNN could be included.

The CERN Summer Student Programme provided me with a tremendous amount of new knowledge and skills. This project has improved my understanding of deep learning and taught me the value of benchmarks. I learned about and used a variety of technologies and tools, which will undoubtedly help me advance in my career and have a beneficial impact on my future.

## **Acknowledgements**

I would like to thank my supervisor, Lorenzo Moneta, for all of his help with this project and for answering any questions I had. I am grateful for all of the new things I have learned at CERN as a Summer Student.