



Contribution ID: 291

Type: Talk

## Real-time implementation of Artificial Intelligence compression algorithm for High-Speed Streaming Readout signals.

Thursday 24 October 2024 17:09 (18 minutes)

The new generation of high-energy physics experiments plans to acquire data in streaming mode. With this approach, it is possible to access the information of the whole detector (organized in time slices) for optimal and lossless triggering of data acquisition. Each front-end channel sends data to the processing node via TCP/IP when an event is detected. The data rate in large detectors is often very high and the network is likely to be a bottleneck for the system. On the other hand, the network devices do not need to know the signal shape but only the hit timing (time-stamp) and the address of the front end that generated it. This is a key point to implement a compression algorithm: it is possible to compress signal samples after the front end and decompress them when data are needed for high-level analysis.

To achieve a high compression ratio and fast inference time on hardware an AI-supported algorithm, an autoencoder, is chosen. Autoencoder is an unsupervised machine learning algorithm composed of two parts: an encoder and a decoder that reduces the size of the input and reconstructs the original input from the encoded representation, respectively.

This contribution describes the compression algorithm and the Streaming Readout DAQ (SRO) system prototype developed to test it. The SRO prototype is designed with three separate nodes connected to the same network. We use a PC as a proxy of the final high-level analysis node and two Raspberry Pi single-board computers as signal generators and data processing units (compressors). The architecture was made such that each node could/will be easily replaced with faster hardware (eg. an FPGA).

Consideration concerning the compression algorithm complexity, loss during compression, and execution time are taken into account to achieve the best tradeoff. Results of autoencoder training and timing of some implemented configurations are reported.

**Authors:** ROSSI, Fabio; Prof. BATTAGLIERI, Marco (INFN); Dr RAGUSA, Edoardo (UNIGE); Prof. GASTALDO, Paolo (UNIGE); GAVALIAN, Gagik (Jefferson Lab)

**Presenter:** ROSSI, Fabio

**Session Classification:** Parallel (Track 2)

**Track Classification:** Track 2 - Online and real-time computing