ROOT MATH

DRAFT WORK PLAN FOR 2018

- Machine Learning (TMVA)
- Math Libraries
- Histograms, Fitting

MACHINE LEARNING

WHY IS IMPORTANT TO HAVE OUR OWN TOOLS

- Machine Learning is becoming more and more important in the HEP analysis eco-systems
  - tools can be applied to a large variety of problems
- Relying completely in software tools which are externals can be risky
  - need to understand and customise tools according to our requirements
  - need to control software development cycles
  - external tools might evolve too fast (change quickly API, becoming obsolete, etc…)
- No need to re-invent the wheel, but develop from experience gained in the Data Science community
- A lots of the effort can be provided by excellent students at a very low man-power cost
- ML tools + efficient I/O can be a very good reason for using ROOT outside HEP (knowledge transfer)
- **Important to have a rich set of modern ML tools in ROOT (e.g. DL) and at the same time provide interfaces to popular external libraries (e.g. Keras, Tensorflow)**

MACHINE LEARNING

- **Improvement in data input**
  - re-design the input TMVA Event data class
  - provide support for Images and Tensors (2-d and multi-dim arrays)
  - automatic conversions to NumPy arrays for using external ML Python tools
  - replace TTreeFormula's with DataFrame
- **Deep Learning Framework**
  - continue development initiated with GSOC students
  - CNN integration is almost done
  - integrate now RNN (and extend to LSTM) and DAE
  - improve performances by optimise code for multi-core CPU and GPU
  - integrate usage of external tools (e.g.)
  - A rich set of tools for DL in HEP

MACHINE LEARNING

(PART 2)

- **Interfaces to external tools**
  - Improve Scikit-learn and Keras interfaces
  - Add a low-level Tensorflow interface
  - All interfaces should provide:
    - train and evaluate in TMVA
    - train externally and evaluate only in TMVA
      (useful for not having external dependency in evaluation)
- **Performance optimisations and parallelisation**
  - optimization of BDT using internal parallelism (MT)
  - optimization of DL for both CPU and GPU
  - fully support parallelization (using multi-process and in clusters) at higher level (e.g. cross validation and hyper-parameter tuning)
  - optimize both training but also model evaluation

MACHINE LEARNING

(PART 3)

- **R&D**
  - Experiment with new algorithms
    - Improve optimisers for DL. Now using just SGD.
    - Multi-target regression which could be used for fast simulation
    - Add tools for unsupervised learning in TMVA (e.g. DAE)

- **Documentation**
  - Improve and update Users Guide for new DL tools
  - Declare obsolete Sourceforge web site
  - Make new online doc available in Doxygen

MATH LIBRARIES

- **Parallelisation of Math Libraries**
  - Fully deploy MT parallelisation in ROOT fitting
  - Math Vectorisation
    - make some important Math functions (e.g. TMath::Gaus) template to support scalar and vector types with VecCore
    - implement basic Math functions in VecCore
    - Vectorisation of numerical integration and differentiation
    - Investigate deploying different Math libraries for the different architectures
- **Random Number Generators**
  - Develop a new common Random number library to be used by both ROOT and Geant
    - with functionality for parallelism (independent streams and fast seeding)
    - with vectorised engines
    - with state of the art generators
      - aim is to have MIXMAX as one of the major generator

HISTOGRAM/FITTING PLAN

- Provide support to users and add new functionality when needed
- Parallelisation of most CPU consuming functions in histograms
  - e.g. Histogram operations like additions (merging)
- Maintain ROOT 7 histogram as a prototype and try to get feedback for its evolution

- R&D
  - Extend capabilities of ROOT fitting (support multi-histogram fits)
    - Study parallelisations in these more un-balanced models
  - Prototype interface to perform fitting in ROOT from simple workspace models
  - Prototype new back-end solution for RooFit
    - e.g. a Tensorflow based implementation ?
  - Investigate GooFit for GPU fitting
  - Autodifferentiation