INL Workshop CERN 2017

Summary of Group Discussion:

External and internal ML Tools

Authors: Claire David, Hans Pabst

The problem

- TMVA is the unique framework available
- Workforce fluctuate over time, difficult to maintain all packages

The situation

- People are moving away from ROOT
- People convert in an out (from ROOT to external tools, then back into ROOT for publishing and archiving policies in HEP)

HEP tools vs external tools

- In HEP our data format (ROOT) is unique. Not seen anywhere else. Moreover, it's too commonly accepted to hope for a change in the near future.
- But ROOT isn't modular enough (yet) for machine learning.
- If data were in another format (tensorflow, dataframe, HDF…) it would be easier.

- Advantages of the external tools:
    - huge community to offer help (StackOverflow) and support (trainings)
    - given the competition for permanent positions, physicist leave HEP for industry and have to learn new tools. Using common tools will avoid an additional learning curve.
    - moving forward as fast as industry in ML field, increase productivity
- Disadvantages of external tools:
    - danger of deciding on a tool now (tensorflow) but it may loose its trend / become obsolete / go into an unadapted direction for ML
    - worse case: no maintenance (unlikely but possible)
    - is it easy to contribute? Are HEP-patches actually taken by maintainers?

Advanced solution: **A middleware:**

In this context a double-wrapper:

1) **from HEP tool to the external tool**
2) **from the external tool to the HEP tool (after ML training)**

Caution: this middleware doesn't imply that it hides completely the other layers. Here one needs knoweldge on both inputs and outputs to make a bridge between the HEP tools and external tools.

Requirements

- should be "agnostic":
  - easily scriptable for productivity (e.g. python? or R?)
  - can support several languages
  - format agnostic? (a jet described in a rootfile or slha or … is still a jet)

    Yes but risk of being too asbtract and need then to write more so that features can be used: this defeats the native purpose.
- "We should take the simplest and adapt it to the more complex" (as low as possible for the training part, but high for the use of the result)
- possibility to contact through the alumni groups the HEP physicists who left for industry and get their recommendations on best external ML tools for HEP. Also some experiments have dedicated groups overlooking on how analyses are done.
- Long term key is automation:
  - avoid intermediary steps (and files)
  - make results reproducible for both experimentalists (PhD students, theorists...)
  - make experiment repeatable in a simpler & faster way

Note: some people used interchangeably the word middleware and plugin. The latter implies the choice of a given framework with contribution from the HEP developers, but in this current discussion no differentiation is made.

Implementation

- how to avoid (likely very heavy) intermediate data format while converting rootfiles?
- should we contribute or rewrite?

News from TMVA (Lorenzo)

- lots of new development in the past year (e.g. on Deep Neural Network, with even better performance than Theano)
- People learning (instead of only using) the algorithms are the best ones: they have the knowledge to make the tool better. Not throwing things into a black box.
- TMVA is good while dealing with weighted events and systematics

Pro: Central tool maintained by the community: newcomer can easily start and improve the tool
Cons: with HEP tools one needs to be an expert to start contributing (steep learning curve) whereas external libraries are easier to read and use.

Conclusion

- ROOT might become more modular
- The least risky way: opening HEP to more/many alternatives to ROOT in parallel
- Decide the right level of abstraction to make it adaptable for many languages & tools
- Survey what local solutions have been already implemented with their local experts
  - get an overview on the trends
  - make a database of expertise within the HEP community
  - contact HEP people who left for industry and benefit from their double-expertise

  Let’s bring a couple of people doing that? Relatively easy.
- Cope with the risk of obsolete language/tool/package with good archiving methods