TMVA Tutorial

Sergei Gleyzer, Lorenzo Moneta
for the ROOT-TMVA Team

IML Workshop, March 21 2017
Outline

• Status and Overview

• New TMVA Features
  – External Interfaces
  – Deep Learning, Jupyter, Parallelization

• Future Plans and Outlook

• Summary
TMVA

Toolkit for Multivariate Analysis:

• HEP Machine Learning workhorse
• Part of ROOT
• In LHC experiments production
• Easy for beginners, powerful for expert
• New TMVA version released in the ROOT version 6.0.8
New TMVA Features

Modularity, External Interfaces, Updated SVMs
Analyzer Tools: Variable Importance

Deep Learning CPU, GPU
Parallelization with multithreading and GPUs
Analyzer Tools: Cross-Validation, Hyper-Parameter Tuning
Regression Loss Functions
Jupyter: Interactive Training, Visualizations
Unsupervised Learning
Deep Autoencoders
Multi-processing, Spark parallelization

Added beginning of 2016
Added in TMVA ROOT 6.0.8
Upcoming
TMVA Interfaces

Interfaces to External ML Tools

- **RMVA** interface to R
- **PyMVA** interface to scikit-learn
- **KMVA** interface to Keras
  - High-level interface to Theano, TensorFlow deep-learning libraries
Deep Learning

New Deep-Learning Library in TMVA

- GPU support
  - CUDA
  - OpenCL

- Excellent performance and high numerical throughput
Deep Learning

Throughput Comparison

2.7 * Theano

Single precision

batch size = 1024

Excellent throughput compared to Theano on same GPU
Deep Learning

ROC Performance: significant improvements compared to shallow networks and boosted decision trees
Regression

New Regression Features:

Loss functions:
- Huber (default)
- Least Squares
- Absolute Deviation
- Custom Function

Important for regression performance

Higher is better
Cross Validation

New features:

• k-fold cross-validation

• Hyper-parameter tuning
  – Find optimized parameters (SVM, BDT)
TMVA Tutorial

• Run tutorial on notebook
  • use SWAN
    • go to swan.cern.ch

• or running local notebooks
  • root — notebook

If you don’t have CERN account for using SWAN please contact me
Some temporary account can be made available
Starting SWAN

make sure to have selected
the development version

SWAN Customisation

Specify the parameters that will be used to contextualise the container which is created for you. See the online SWAN guide for more details.

Software stack

Development Bleeding Edge (might be unstable)

Platform

x86_64-abc-objc-49-opt

Environment script

e.g. $CERNbox_HOME/MySWAN/my_script.sh

Number of cores

1

Start my Session

click here to start
Starting a Terminal in SWAN

After login cernbox home directory will be visible

Start a terminal window
Getting the Notebooks

• Clone the git repository
  • `git clone https://github.com/lmoneta/tmva-tutorial.git`
  (link available also in Indico)
TMVA Contributors

- Sergei Gleyzer: Analyzer Tools, Algorithm Development
- Lorenzo Moneta: Multi-threading, Multi-processing
- Omar Zapata Mesa: PyMVA, RMVA, Modularity, Parallelization
- Peter Speckmeyer: Deep-Learning CPU
- Simon Pfreundschuh: Deep-Learning CPU and GPU
- Adrian Bevan, Tom Stevenson: SVMs, Cross-Validation, Hyperparameter Tuning
- Attila Bagoly: Jupyter Integration, Visualization, Output
- Albulena Saliji: TMVA Output Transformation
- Stefan Wunsch: KERAS Interface
- Pourya Vakilipourtakalou: Cross-Validation, Parallelization
- Abhinav Moudhil: Pre-processing, Deep Autoencoders
- Georgios Douzas: Spark, Cross-Validation, Hyperparameter Tuning
- Paul Seyfert: Performance optimization of MLP
- Andrew Carnes: Regression, Loss Functions, BDT Parallelization

Continued invaluable contributions from Andreas Hoecker, Helge Voss, Eckhard von Thorne, Jörg Stelzer, and key support from CERN EP-SFT Group
More Information

Websites:  http://root.cern.ch
          http://iml.cern.ch
          http://oproject.org