Machine Learning Developments in ROOT

Sergei Gleyzer, Lorenzo Moneta

for the ROOT-TMVA Team

CHEP 2016, October 10, 2016

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 experts
- 17 active contributors (5 GSoCs)

New TMVA version released in upcoming ROOT 6.0.8

New TMVA Features

New 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 in 2015

Added in TMVA ROOT 6.0.8

Upcoming 2016

TMVA Interfaces

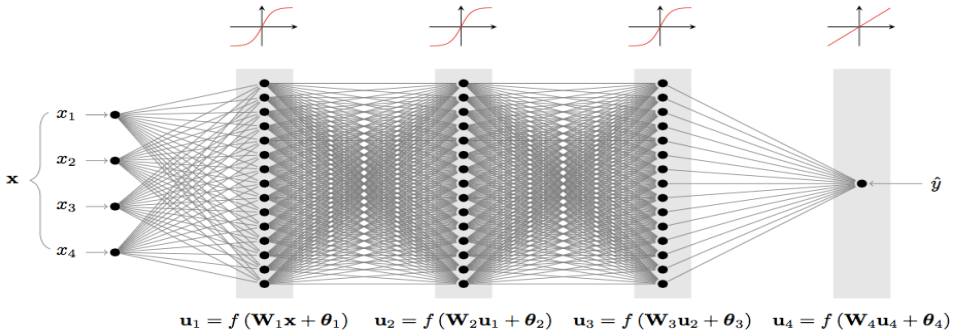
Interfaces to External ML Tools

- **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

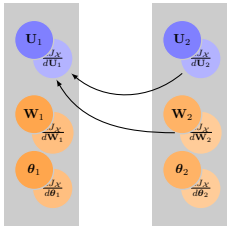
- GPU support
  - CUDA
  - OpenCL

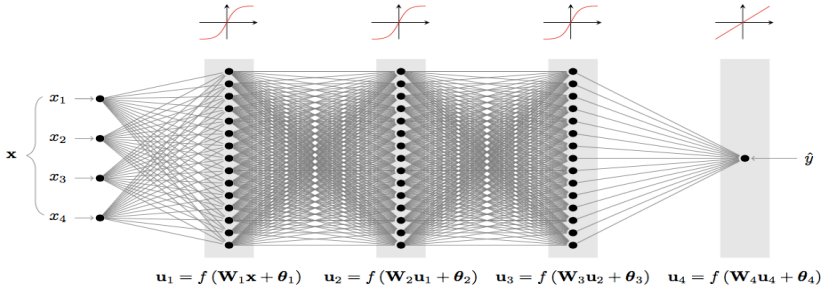


- Excellent performance and high numerical throughput

Deep Learning

Is a powerful Machine Learning method based on Deep Neural Networks (DNN) that achieves significant performance improvement in classification tasks





Deep Learning

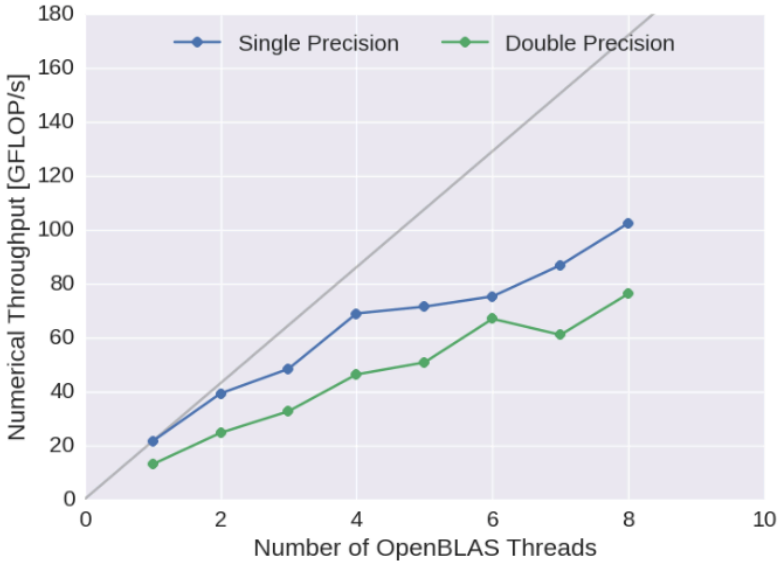
CPU Performance:

Implementation:

- OpenBLAS, TBB

Peak performance per core:

- 16 GFLOP/s
- Single, Double Precision



Deep Learning

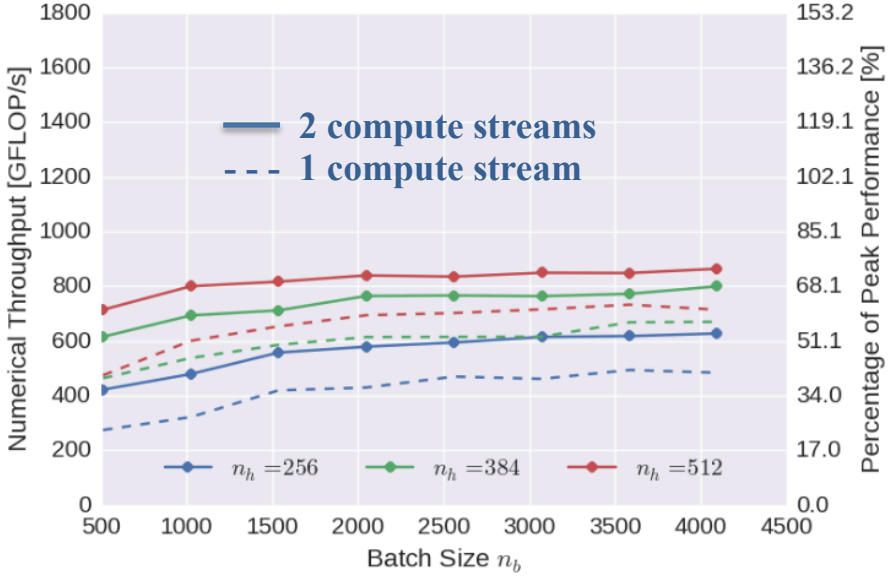
GPU Performance:

Network:

- 20 input nodes
- 5 hidden layers with n_h nodes each

Hardware:

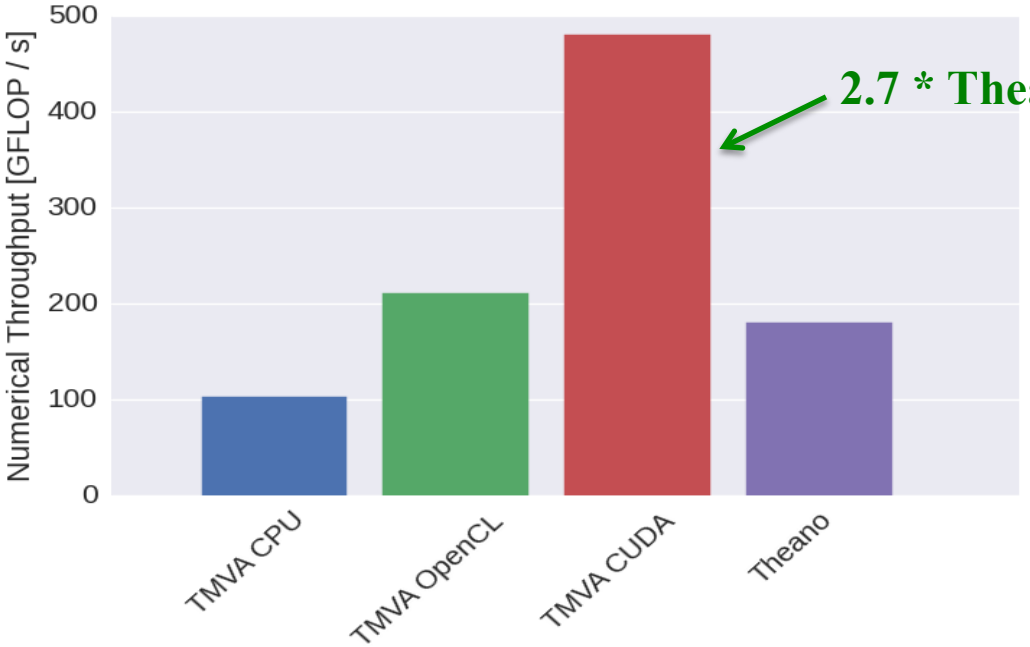
- NVIDIA Tesla K20
- 1.17 TFLOP/s peak performance @ double precision



Good Throughput

Deep Learning

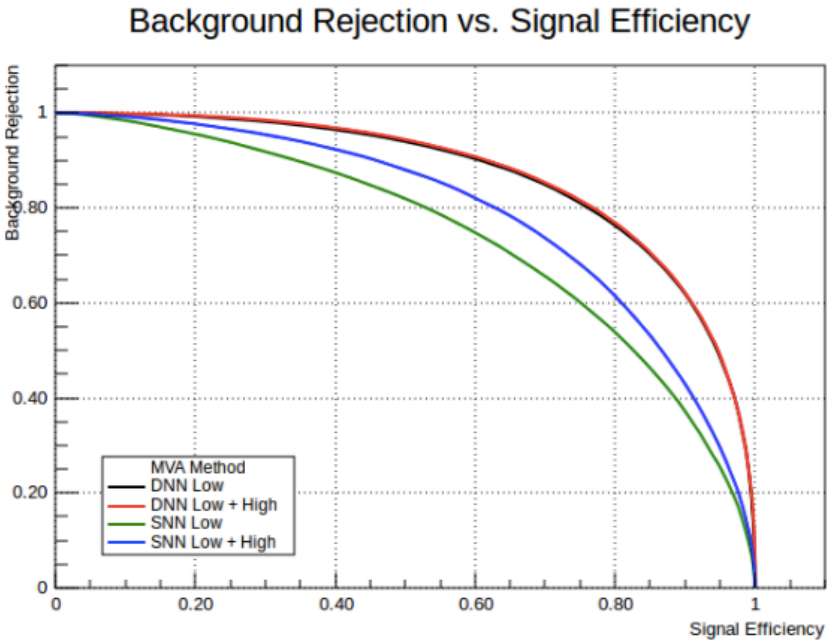
Throughput Comparison

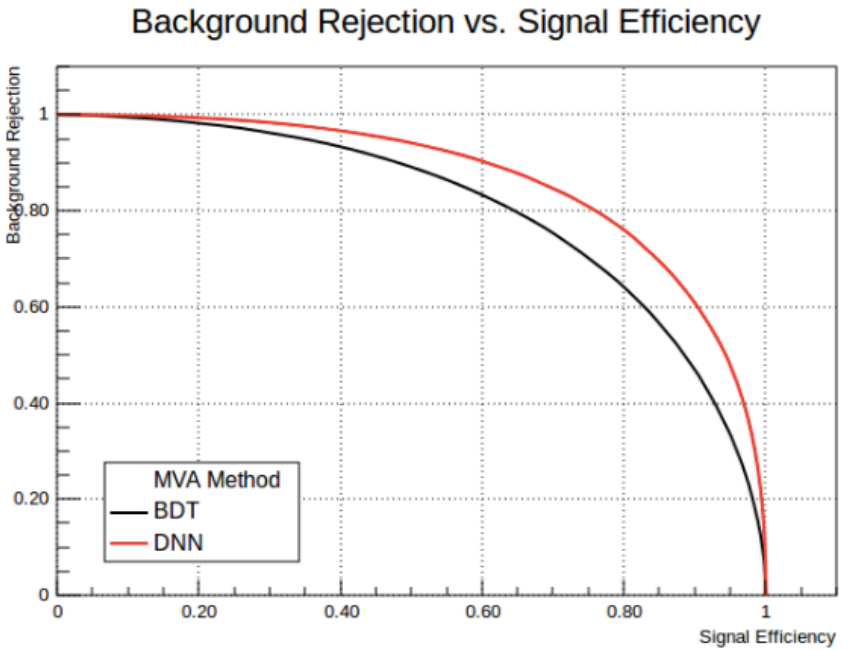


batch size = 1024

Excellent throughput compared to Theano on same GPU

Deep Learning



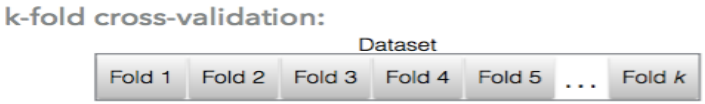


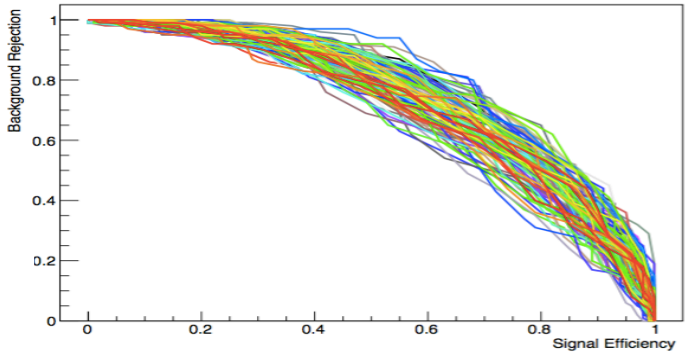
ROC Performance: significant improvements compared to shallow networks and boosted decision trees

Cross Validation

New features:

- **k-fold cross-validation**





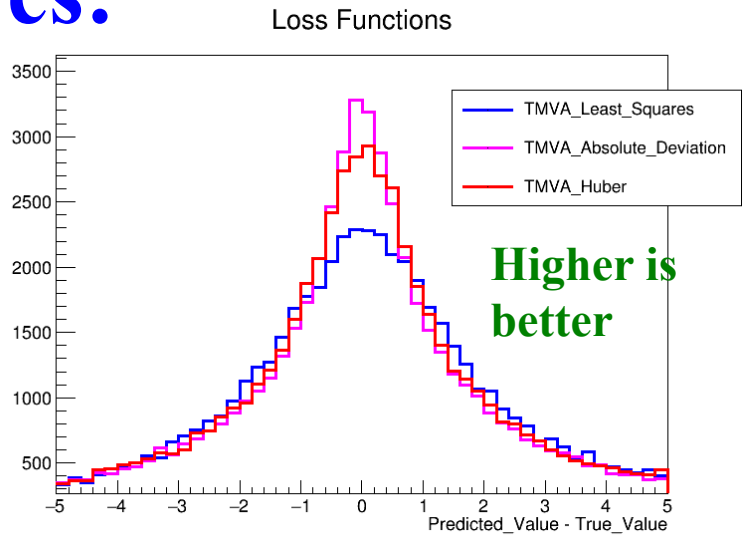
- **Hyper-parameter tuning**
  - **Find optimized parameters (SVM, BDT)**

Regression

New Regression Features:

Loss functions:

- Huber (default)
- Least Squares
- Absolute Deviation
- Custom Function



Important for regression performance

Jupyter Integration

Classifier output: Neural networks, decision trees

Simple neural network

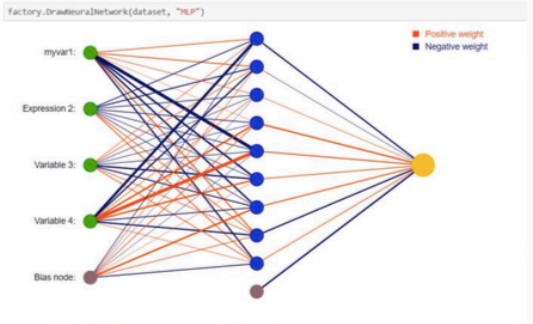
- Python function reads the network, converts to JSON; JS with d3js make the visualization from JSON
- Interactive: focusing connections, zooming, moving

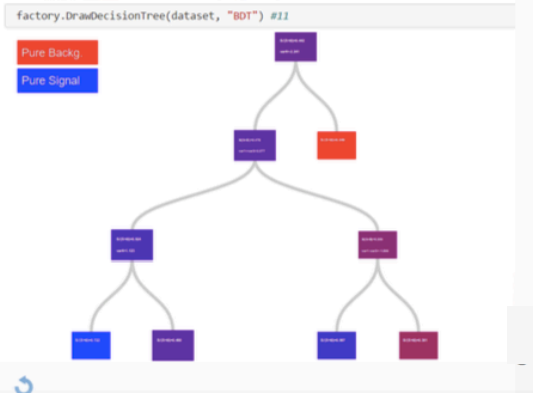
Deep neural network

- HTML5 Canvas visualization (speed)
- Less interactive: zooming, moving

Decision trees

- Ipywidgets: input field for selecting the tree
- Visualization from JSON with D3js
- Interactive: closing subtree, showing the path, focusing, moving, zooming, reset

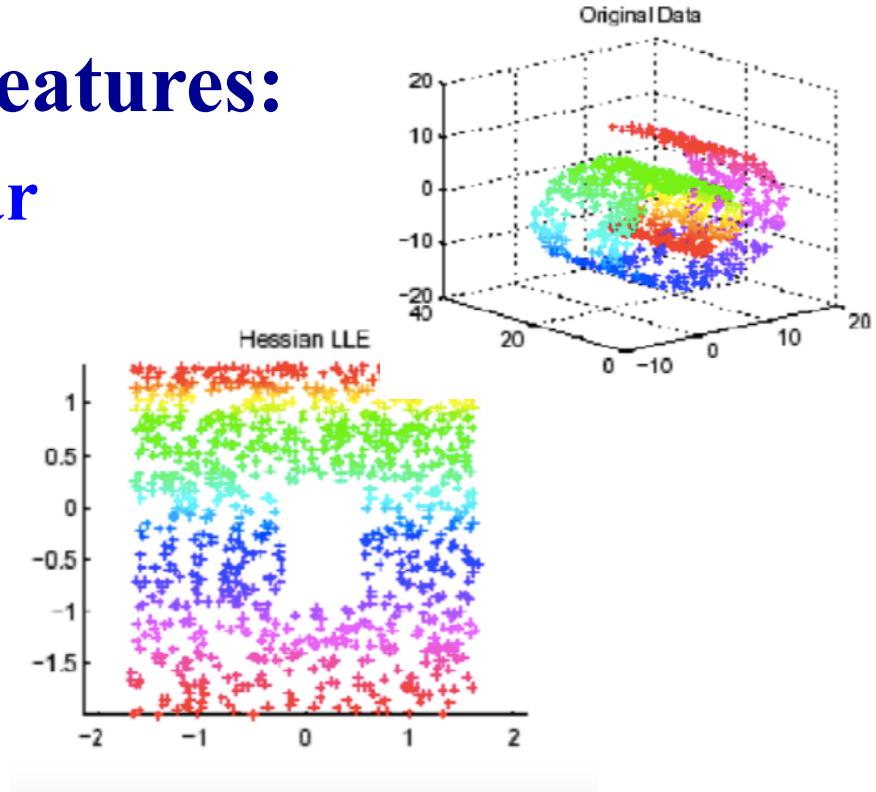




Pre-processing

New pre-processing features:

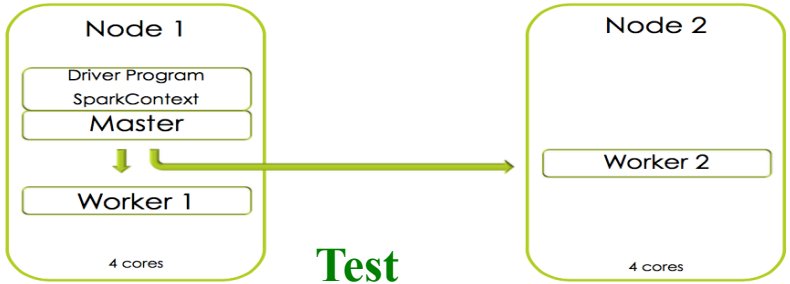
- **Hessian Locally Linear Embedding**
  - **(Hessian LLE)**
- **Variance Threshold**



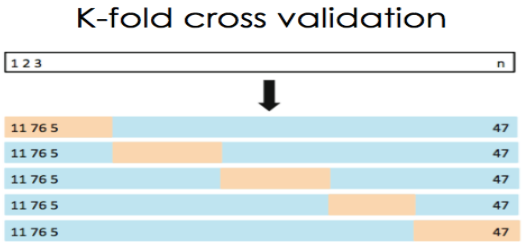
Some Upcoming Features

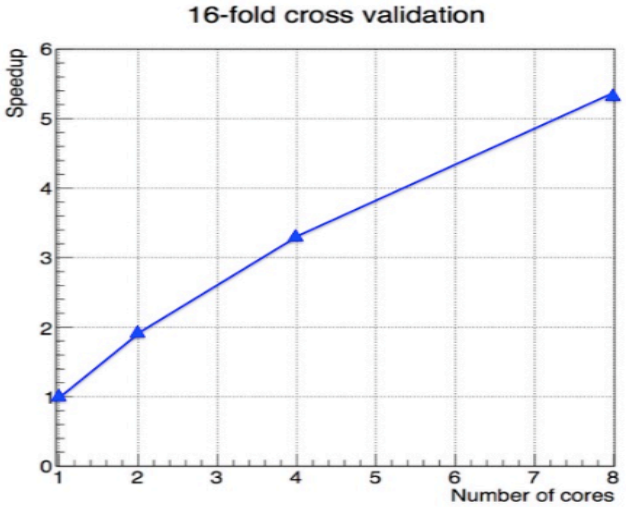
Spark TMVA

SPARK Parallelization



Test

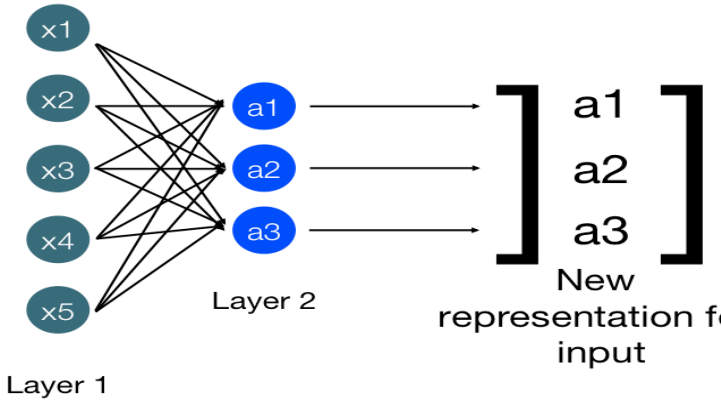




Good speed-up in prototype R&D

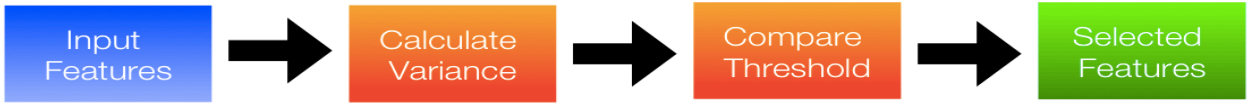
Deep Autoencoder

Deep Autoencoders



- Deep neural network is trained to output the input i.e. learn the identity functions.
- Constrain number of units in hidden layer, thus learning compressed representation.

Variance Threshold



Summary

- **Many new features in TMVA release upcoming in ROOT 6.0.8**
  - **Production-ready parallelized Deep Learning**
  - **Cross-validation, Hyper-parameter tuning**
  - **Jupyter integration**
  - **More pre-processing features**
  - **Regression updates**
- **Many contributions**
- **Feedback and further contributions welcome**

Feature Contributors

Sergei Gleyzer	Analyzer Tools, Algorithm Development
Lorenzo Moneta	Multi-threading, Multi-processing
Omar Zapata Mesa	PyMVA, RMVA, Parallelization
Peter Speckmeyer	Deep-Learning CPU
Simon Pfreundschuh	Deep-Learning GPU
Adrian Bevan, Tom Stevenson	SVMs, Cross-Validation, Hyperparameter Tuning
Attila Bagoly	Jupyter Integration, Visualization, Output
Albulena Saliji	TMVA Output Transformation
Stefan Wunsch	KERAS Interfance
Pourya Vakilipourtakalou	Cross-Validation, Multi-threading
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

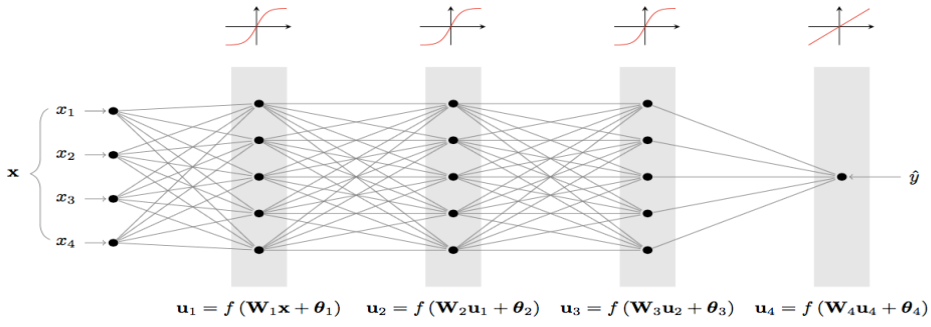
Websites: http://root.cern.ch
http://iml.cern.ch
http://oproject.org

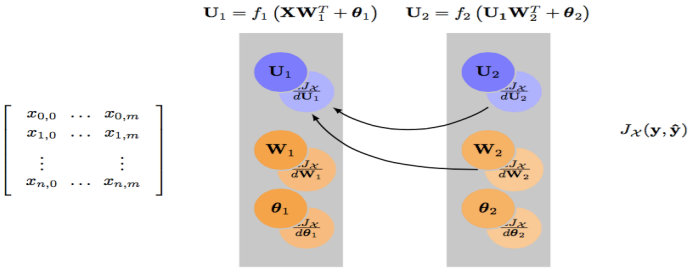
Inter-experimental LHC Machine Learning working group

- Exchange of HEP-ML expertise and experience among LHC experiments
- ML Forum
- ML software development and maintenance
- Exchange between HEP and ML communities
- Education (Tutorials)

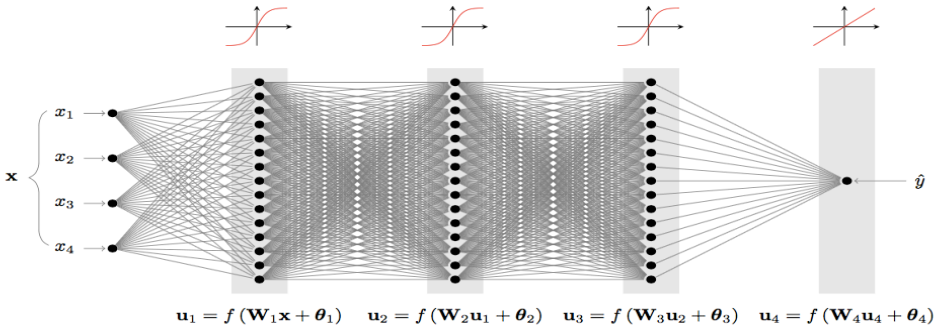
Backup

TMVA Deep Learning





Design



link