Notes from
Heavy Flavour Data Mining Workshop

Eduardo Rodrigues
University of Cincinnati

DIANA Meeting, CERN, 22 February 2016

Heavy Flavour Data Mining workshop

Goal(s)

- Physicists share their challenges and used tools
- ML experts share their experience and tools available
- Provide overview and hands-on experience for popular tools and methods in various fields of Machine Learning

Details

- 1st of its kind in the heavy flavour community (as far as I know)
  - Note: not specifically targeted at heavy flavour community in any way ;-)
- Held in Zurich 18-20 Feb. 2016
- https://indico.cern.ch/event/433556/
- Agenda contained general talks, tutorials and OpenSpace technology discussion sessions

Randon / general remarks

- Very informal $\rightarrow$ facilitated interaction and discussions
- Excellent presentations and tutorials, very interesting / useful

- There is a large gap between what HEP community uses and what is available
- Also the language of both communities are often not the same

  $\Rightarrow$ we need to catch up !

- Whole world out there with techniques/tools often un-/under-used in HEP
  - E.g. scikit-learn and deep learning tools
- Typically used via Python

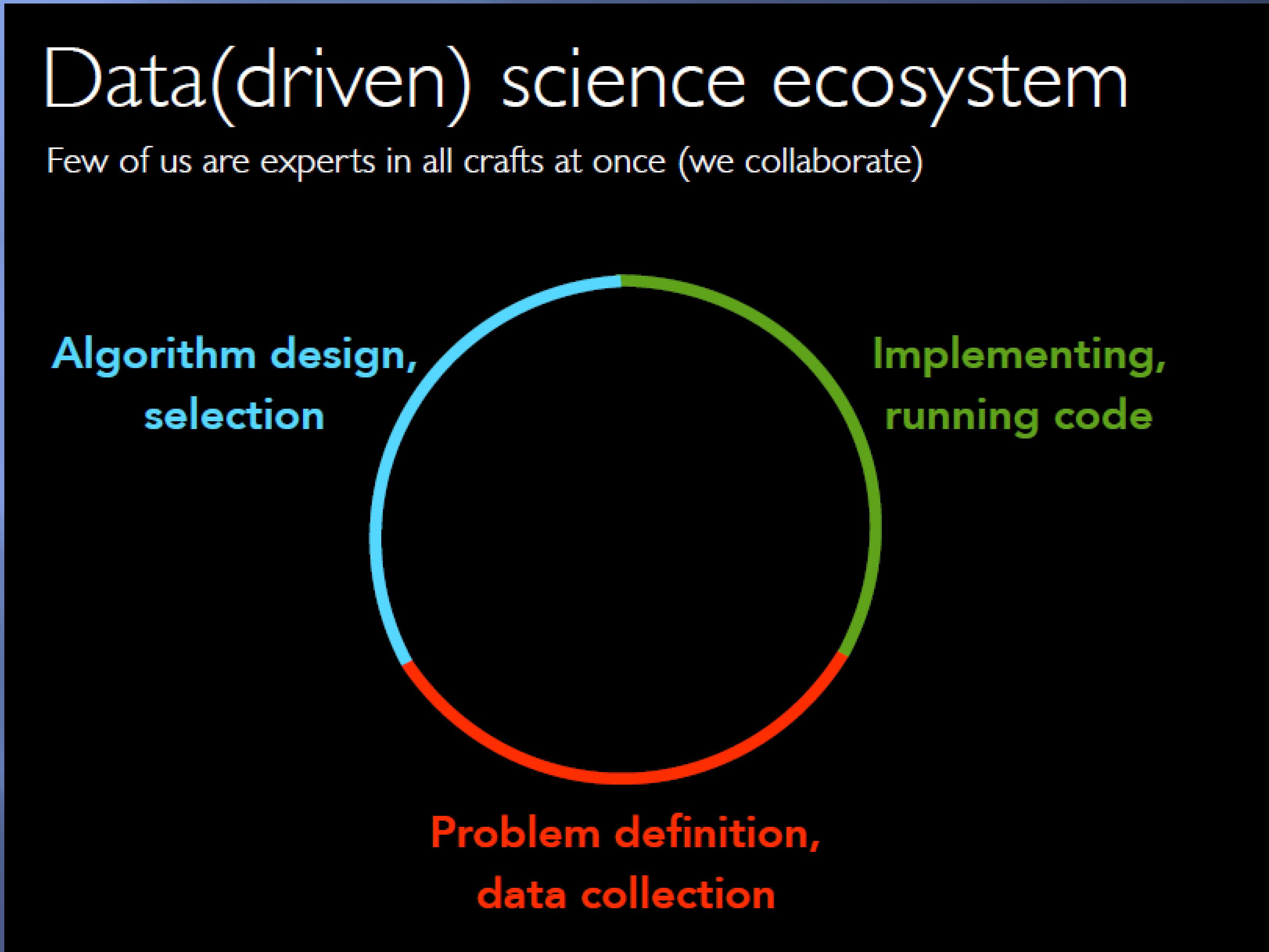
  $\Rightarrow$ again we need to catch up !

OpenML – collaborative ML

Joaquin Vanschoren

OpenML – collaborative ML

Joaquin Vanschoren



Overview of topics presented

- Introduction to challenges in HEP relevant to ML techniques
- Pitfalls of evaluating a classifier's performance in HEP applications

- Mathematics of Big Data
- Decision trees
- Non-trivial applications of boosting (e.g. reweighting distributions)
- Transfer learning
- Data doping
- Tuning of hyper-parameters
- Classifier output calibration
- Data fusion
- Formal Concept Analysis & Concept-based clustering
- Multi-label learning
- Deep learning
- Latent variable modelling

- Etc.

Overview of software projects presented

- hep_ml, scikit-learn, TMVA, XGBoost, etc.

- Jupyter
- Github
- OpenML
- Crowdsourcing

- Reproducible Experiment Platform (REP) & Everware

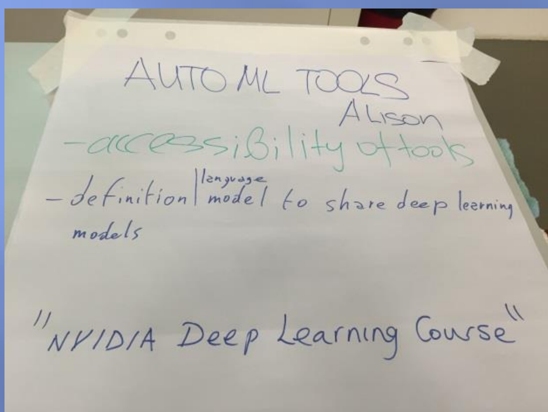
- OpenML

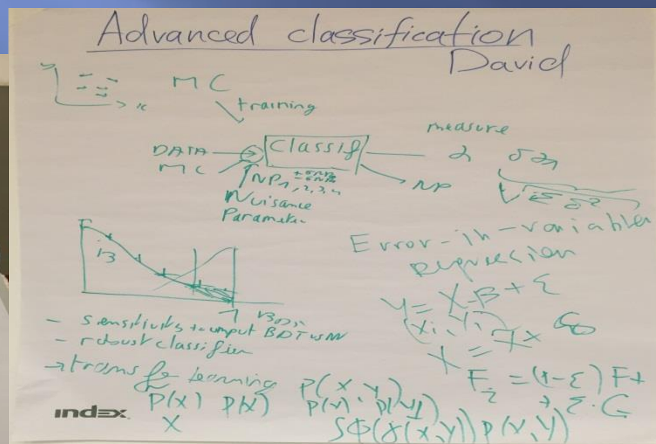
- TensorFlow
- Theano
- Keras

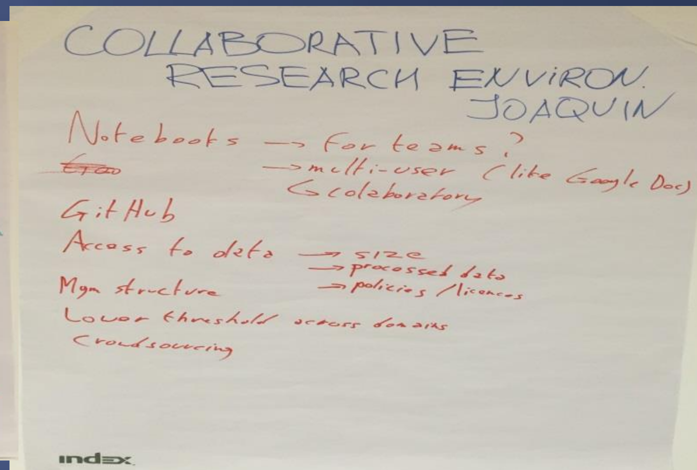
- Etc.

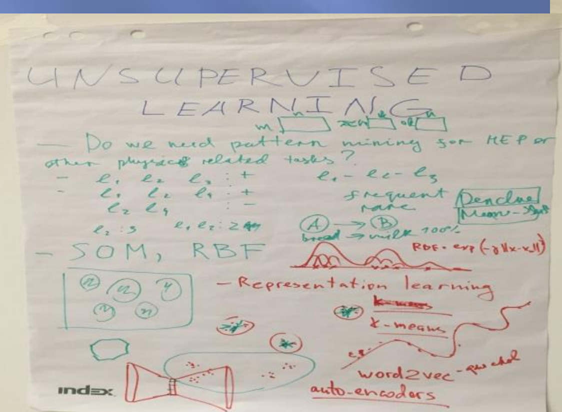
OpenSpace discussions

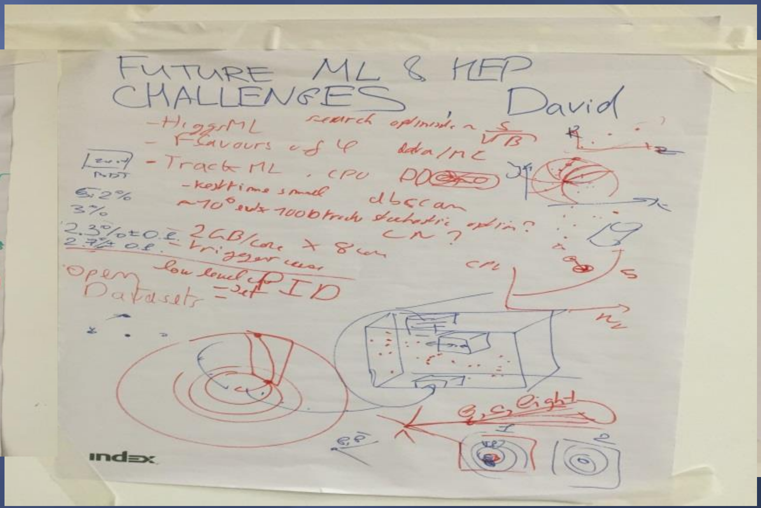
- HEP-specific cases of ML
- Future ML & HEP challenges
- Advanced classification
- Collaborative research environment
- Infrastructure optimisation
- Anomaly detection
- Etc.

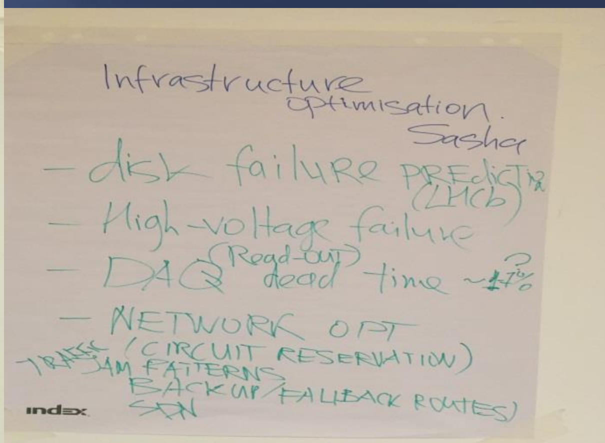












Transfer learning

Alexander Rakhlin

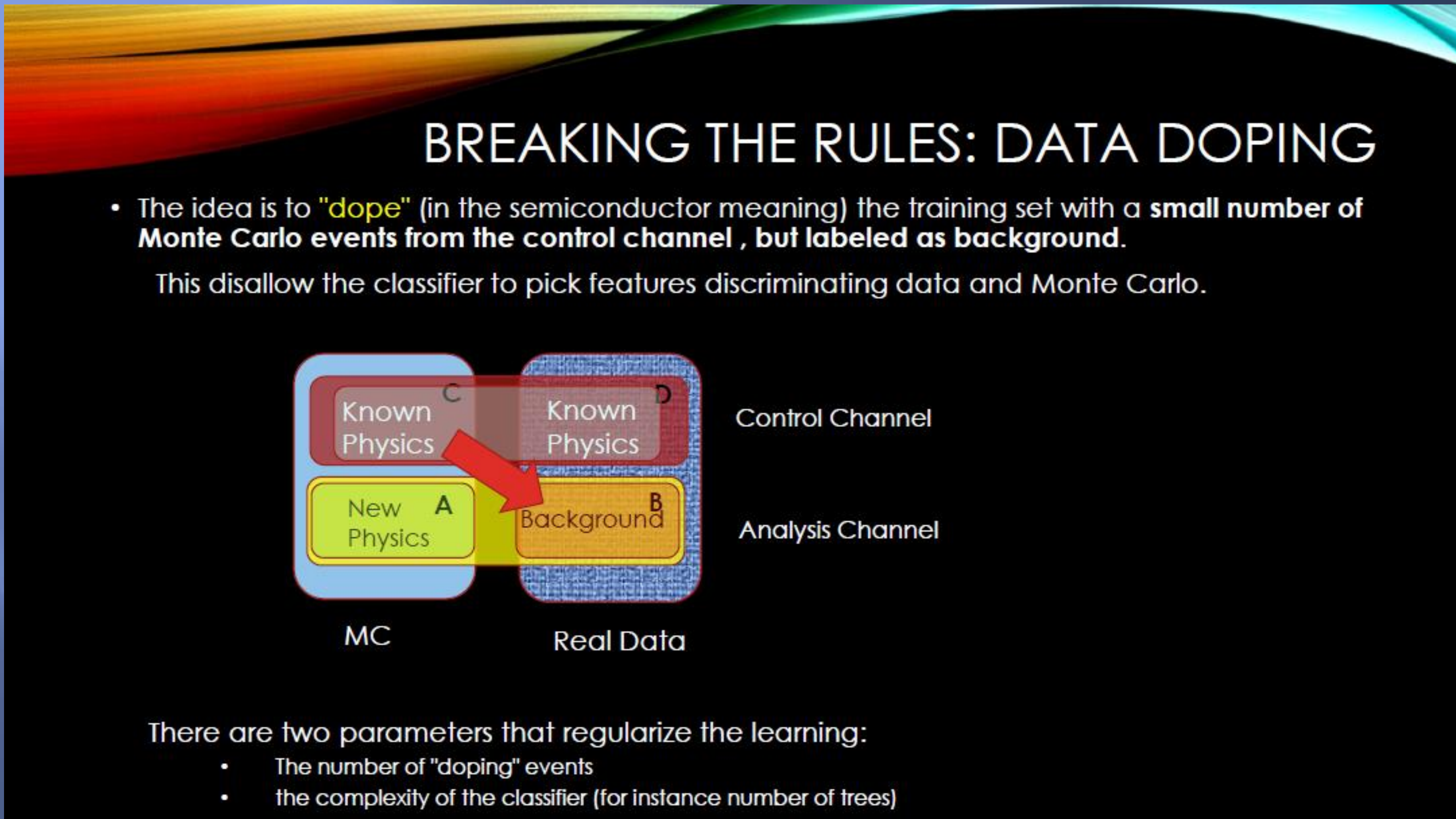
- **Winner of physics prize of Kaggle challenge "Flavours of Physics Challenge" used transfer learning to devise his algorithm**

> The idea of attesting model on control channel is certainly reasonable and can be implemented in theoretically sound way. In Machine Learning the problem of different data sets is well known, and solution is called **Transfer learning.** It aims at transferring knowledge from a model created on the train data set to the test data set, assuming they differ in some aspects, e.g. in distribution.

- **Can we use this to account for data-MC differences … ?**

Data doping

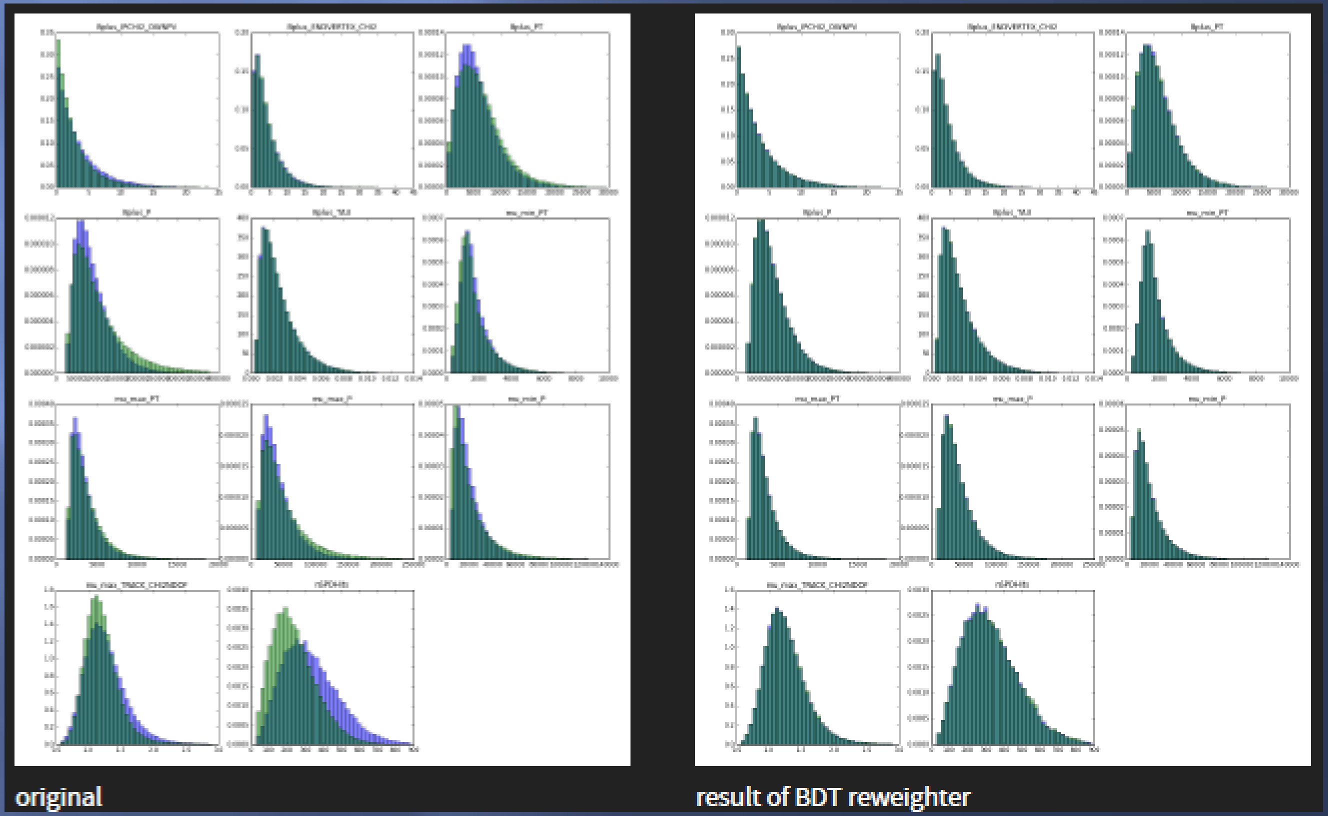
- **Approach used by another winner of physics prize of Kaggle challenge**



Reweighting dists with boosting

Alex Rogozhnikov

- Goal: assign weights to MC such that it matches data distribution
- Trivial in 1D but much harder job if reweighting in various dimensions
- “BDT reweighter” implemented in package hep_ml



Tuning of hyper-parameters

Alexander Fonarev
Artem Vorozhtsov

- **Hyper-parameters examples:**
  tree depth in decision trees, gradient descent step size in NNs, etc.

Open source implementations

Package	License	URL	Language	Model
SMAC	Academic non-commercial license.	http://www.cs.ubc.ca/labs/beta/Projects/SMAC	Java	Random forest
Hyperopt	BSD	https://github.com/hyperopt/hyperopt	Python	Tree Parzen estimator
Spearmint	Academic non-commercial license.	https://github.com/HIPS/Spearmint	Python	Gaussian process
Bayesopt	GPL	http://rmcantin.bitbucket.org/html	C++	Gaussian process
PyBO	BSD	https://github.com/mwhoffman/pybo	Python	Gaussian process
MOE	Apache 2.0	https://github.com/Yelp/MOE	Python / C++	Gaussian process

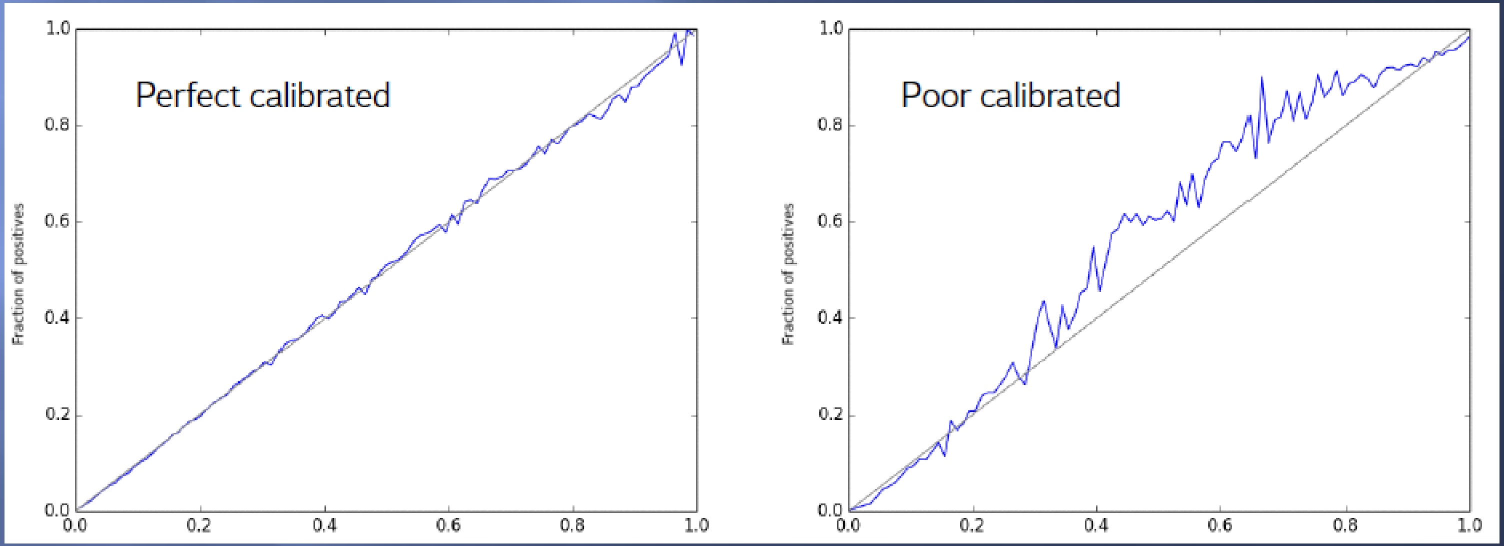
Hutter, Frank, Jörg Lücke, and Lars Schmidt-Thieme. "Beyond Manual Tuning of Hyperparameters." 2015.

Classifier output calibration

Tatiana Likhomanenko

- We often need the output value to be a true probability, from 0 to 1 – obvious
- But not all classifiers are probabilistic, e.g. Support Vector Machines

- Classification = transformation of the score returned by a classifier into a
- posterior class probability



Calibration methods discussed

- Quantile binning, Platt scaling, isotonic regression

Data Fusion

Eugene Burnaev

Problem statement

- construct approximation of high-fidelity function

Data Fusion

Two types of data source with different fidelities are given

Low fidelity function $f_l(x)$	High fidelity function $f_h(x)$
• Cheaper, but less accurate • Bigger database • Better design domain cover Data source examples: • CFD code with coarse mesh • Full-potential equations solver • Numerical simulations	• Accurate, but more expensive • Smaller database • Worse design domain cover Data source examples: • CFD code with tight mesh • Euler equations solver • Real-world simulations

Various methods discussed

- Difference approximation, co-kriging, etc.
- Tensor product of approximations

Deep Learning

Why go Deep?

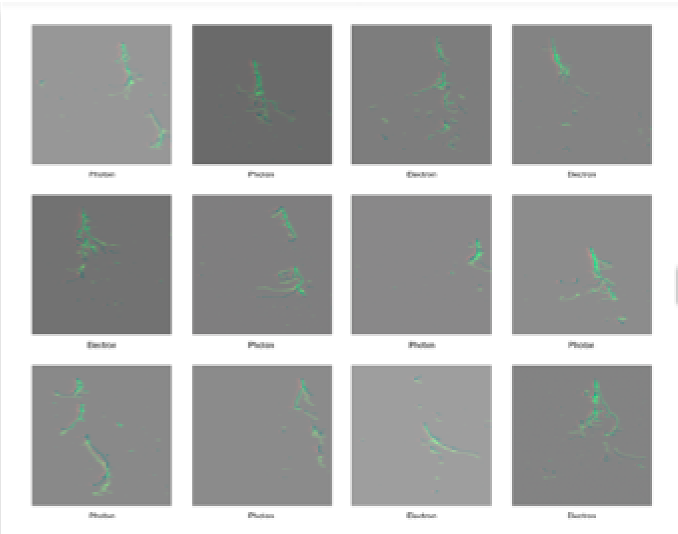
- Eliminate *Feature Engineering*
  - Shallow networks, most of *your* time spent on developing algorithms that process raw data into the inputs (i.e. Reconstruction) to the NN.
  - Deep NNs can learn features from raw data.
- *Parallelization*: DNN are likely faster than traditional algorithms and ideal for GPUs, HPC, …
- *Unsupervised learning*: DNNs classify events without being told what are the classes.
  - The hope is that DNNs could make sense of complicated data that we don't understand.

Amir Farbin

- **One presentation with application to reconstruction in HEP**
- **TensorFlow intro & tutorial**

Great expectations – DL event reco

Amir Farbin

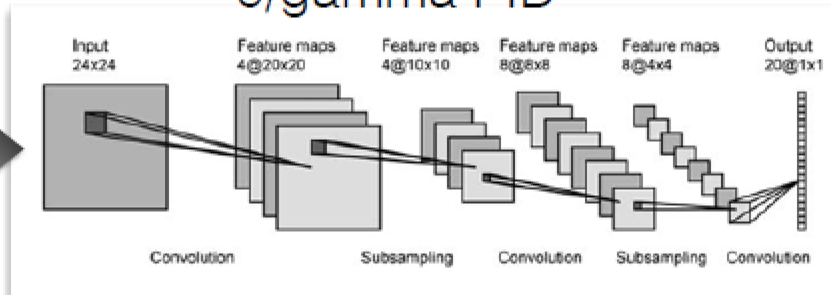


Raw Data: Wire ADC vs Time x Planes
(LArIAT Simulation)

Train

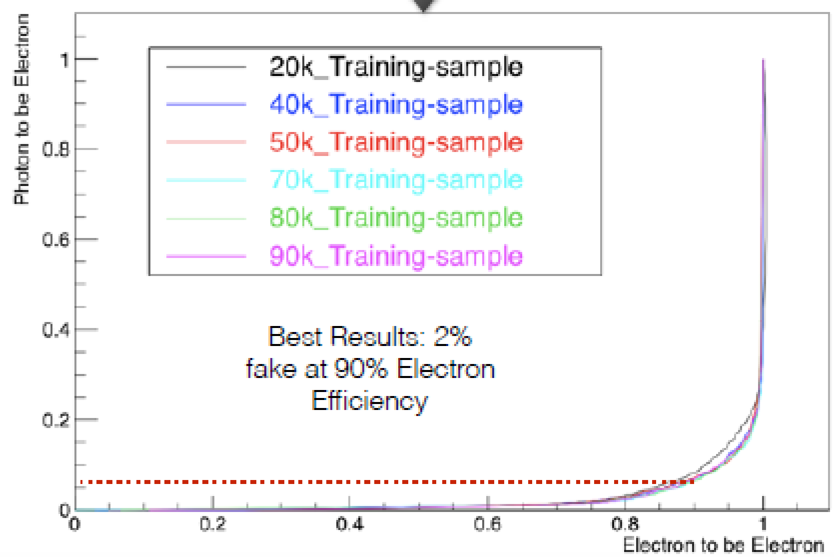
Deep Learning

e/gamma PID



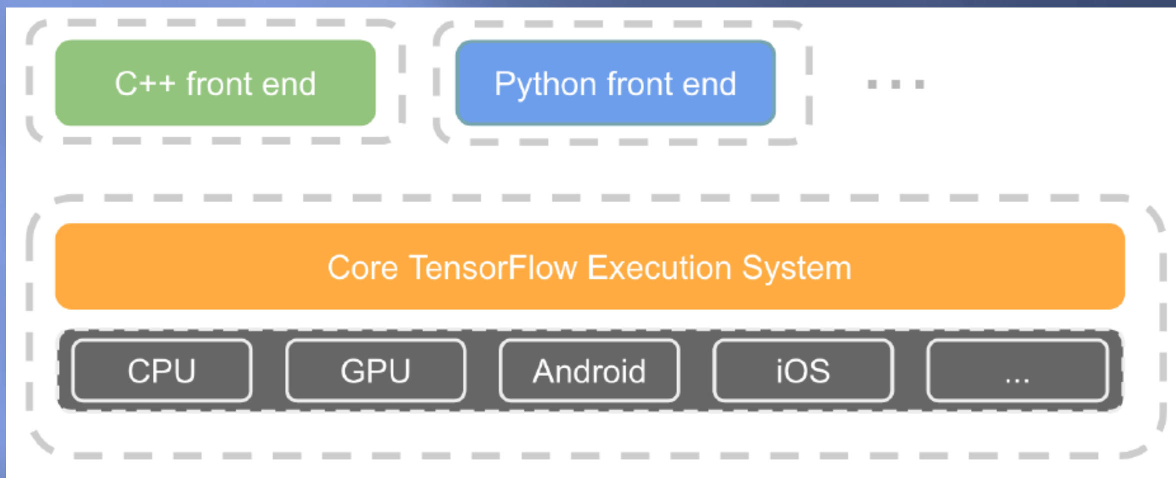
Deep Convolutional Neutral Network
(GoogLeNet)

Out of the box *Feasibility Study* with No attempt at optimization.



TensorFlow

- **Open source software library for numerical computation using data flow charts**



What is a Data Flow Graph?

Data flow graphs describe mathematical computation with a directed graph of nodes & edges. Nodes typically implement mathematical operations, but can also represent endpoints to feed in data, push out results, or read/write persistent variables. Edges describe the input/output relationships between nodes. These data edges carry dynamically-sized multidimensional data arrays, or tensors. The flow of tensors through the graph is where TensorFlow gets its name. Nodes are assigned to computational devices and execute asynchronously and in parallel once all the tensors on their incoming edges becomes available.