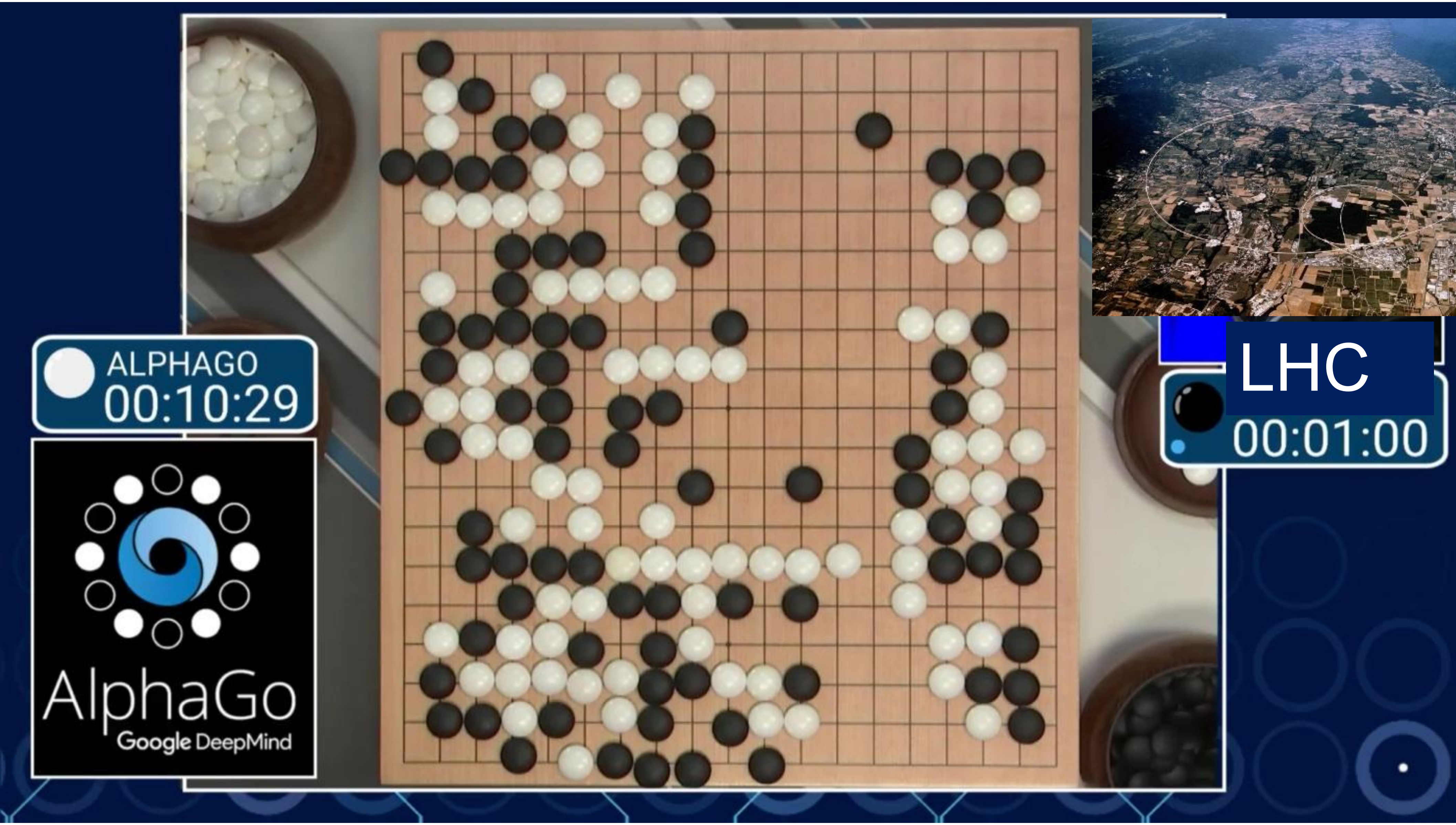
Machine learning at LHC



Dr. Leonid Serkin (ICTP/Udine/CERN)

In "Nature" 27 January 2016:

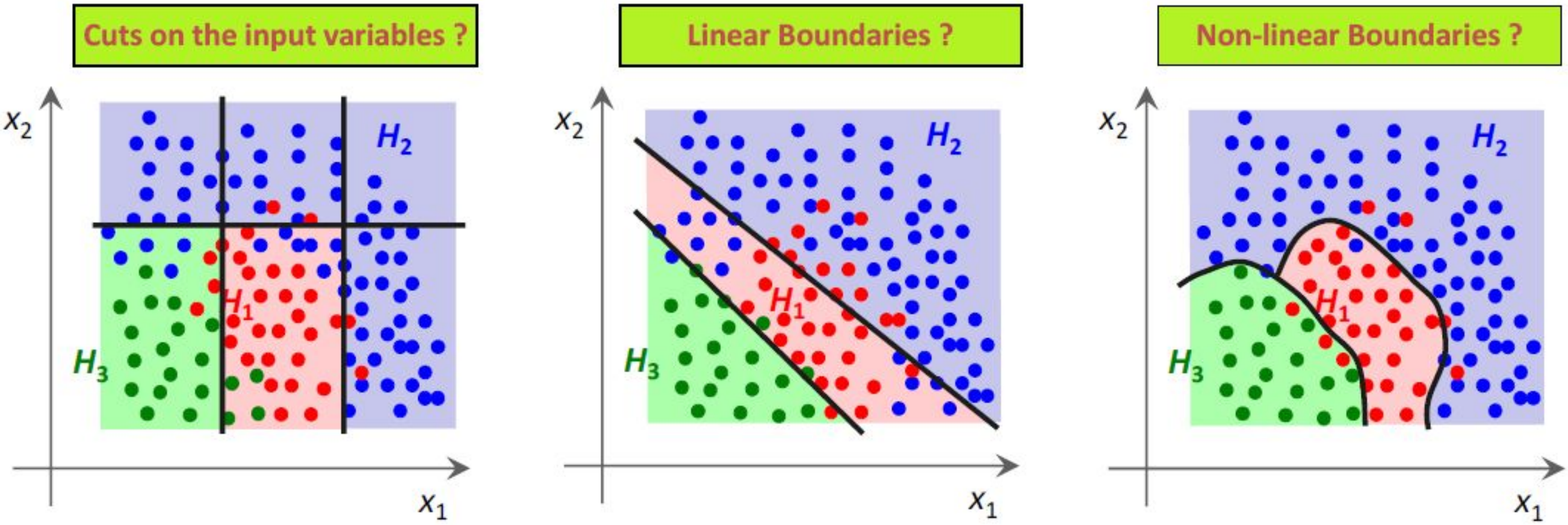
"DeepMind's program AlphaGo beat Fan Hui, the European Go champion, five times out of five in tournament conditions..."

"AlphaGo **was not preprogrammed to play Go**: rather, it learned using a general-purpose algorithm that allowed it to interpret the game's patterns."

"…AlphaGo uses a **Monte Carlo** tree search algorithm to find its moves based on knowledge previously "learned" by machine learning, specifically by an **artificial neural network** (a deep learning method) by extensive training, both from human and computer play

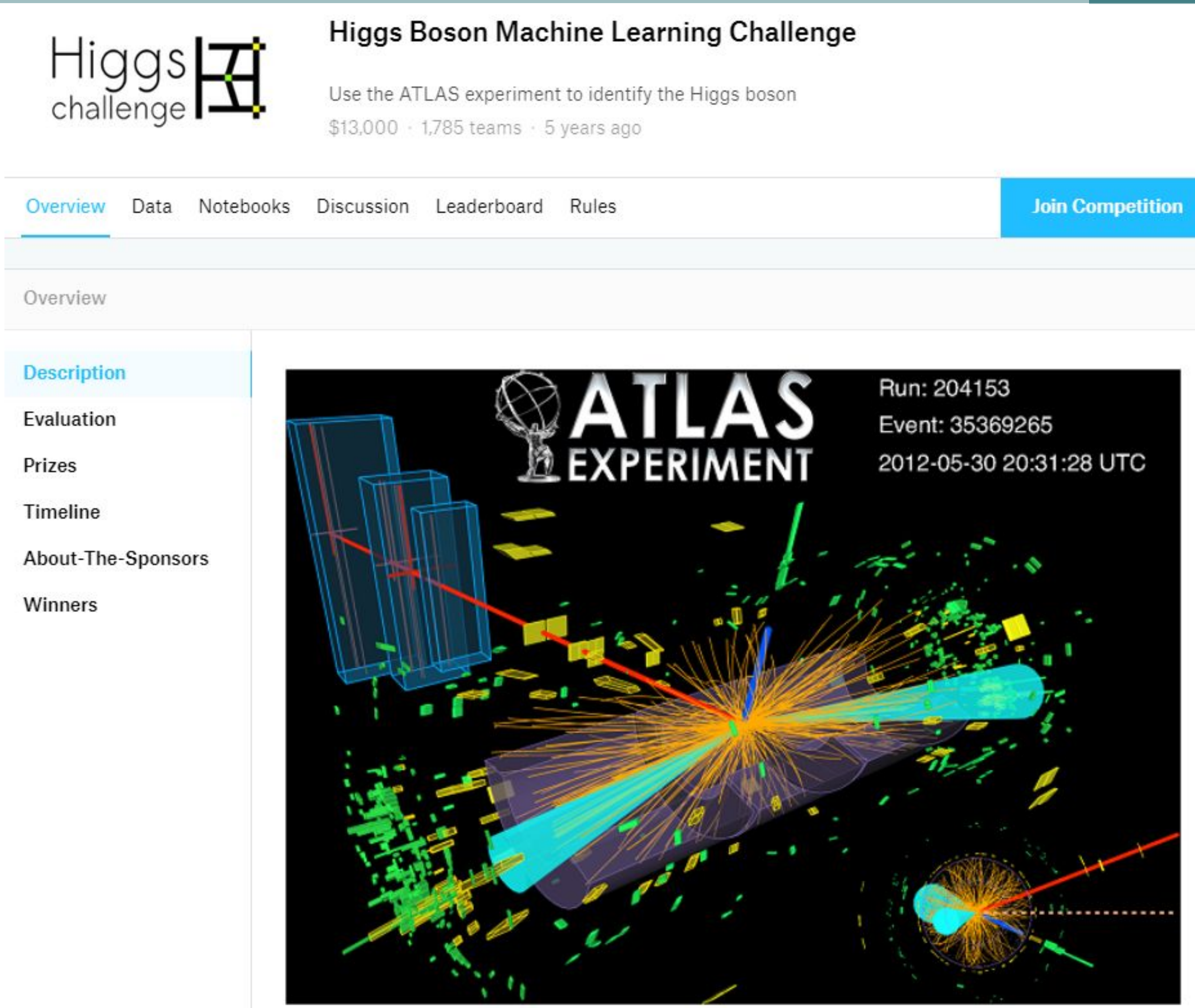
Event classification problem (applied to HEP)

The question: what 'decision boundary' should we use to accept/reject events as belonging to event types *H1, H2 or H3?*



Methods available (up to 2015): Rectangular cut optimization, Projective likelihood estimation, Multidimensional probability density estimation, Multidimensional k-nearest neighbor classifier, Linear discriminant analysis (H-Matrix and Fisher discriminants), Function discriminant analysis, Predictive learning via rule ensembles, Support Vector Machines, Artificial neural networks, Boosted/Bagged decision trees (BDT)…

Higgs Boson ML Challenge



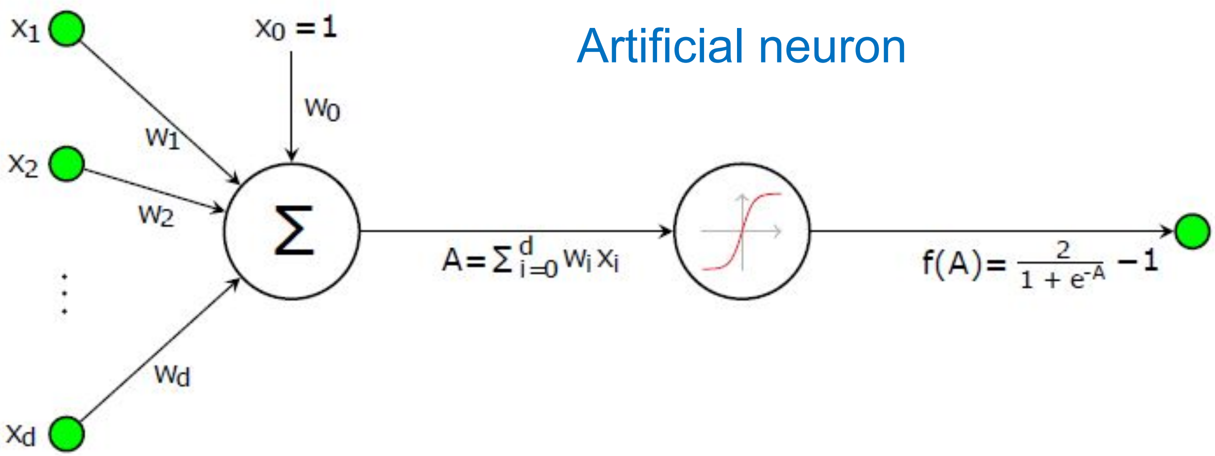
The Higgs Boson Machine Learning Challenge was organized to promote collaboration between high energy physicists and data scientists. The ATLAS experiment at CERN provided simulated data that has been used by physicists in a search for the Higgs boson.

Rank	Method
1	DNN
2	RGF and meta ensemble
3	Ensemble of neural networks
8	XGboost and Intensive feature engineering
31	Ensemble of cascades and non-cascaded models
45	XGBoost Tuned
782	TMVA Tuned
902	MultiBoost
991	TMVA

https://www.kaggle.com/c/higgs-boson

https://higgsml.lal.in2p3.fr/

Typical neural network circa 2005



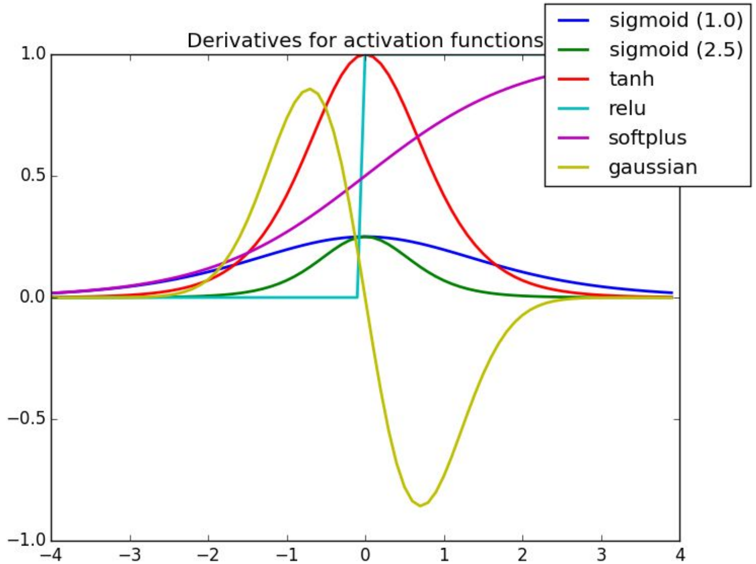
Artificial neuron

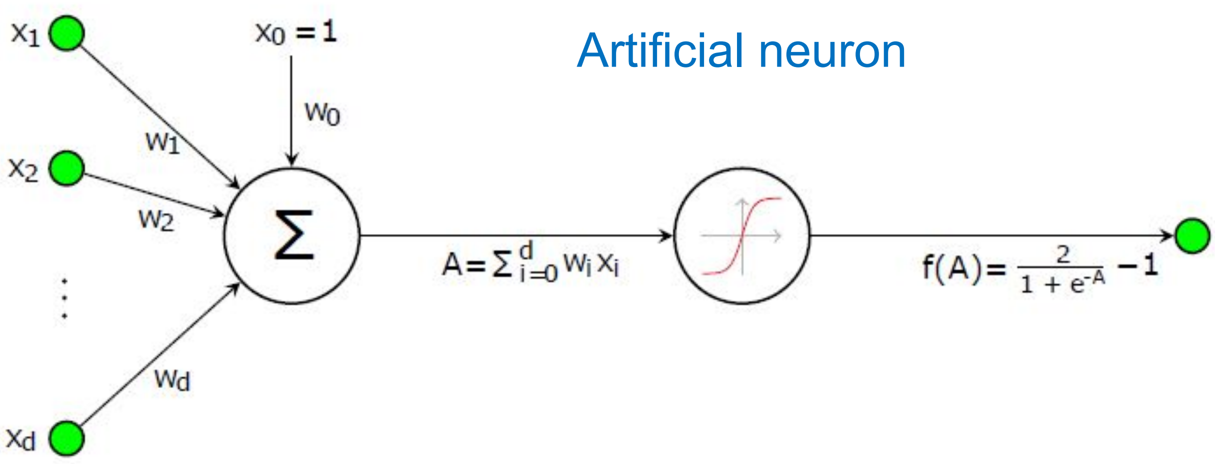
An ANN mimics the behaviour of the biological neuronal networks and consists of an interconnected group of processing elements (referred to as neurons or nodes) arranged in layers.

The first layer, known as the input layer, receives the input variables (x1; x2; …xd). Each connection to the neuron is characterised by a weight (w1; w2; … wd) which can be excitatory (positive weight) or inhibitory (negative weight). Moreover, each layer may have a bias (x0 = 1), which can provide a constant shift to the total neuronal input net activation (A), in this case a sigmoid function:

$$A = \sum_{i=1}^{d} w_i x_i + w_0 = \sum_{i=0}^{d} w_i x_i.$$

$$f(A) = \frac{2}{1+e^{-A}} - 1,$$





The last layer represents the final response of the ANN, which in the case of d input variables and nH nodes in the hidden layer can be expressed as:

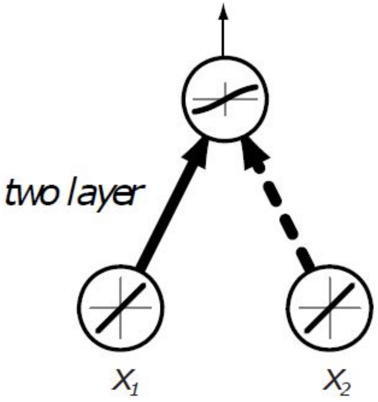
$$o = f\left(\sum_{j=0}^{n_H} w_j f(\sum_{i=0}^{d} w_i x_i)\right)$$

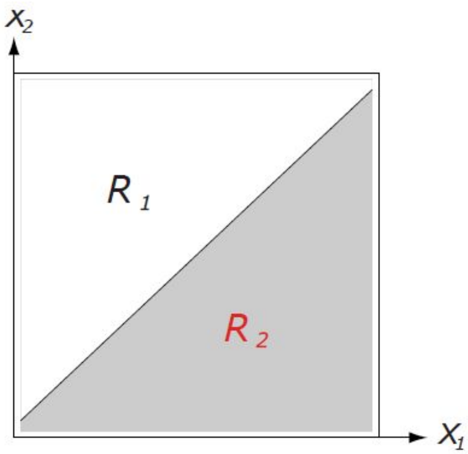
The weights and thresholds are the network parameters, whose values are learned during the training phase by looping through the training data several hundreds of times. These parameters are determined by minimising an empirical loss function over all the events N in the training sample and adjusting the weights iteratively in the multidimensional space, such that the deviation E of the actual network output o from the desired (target) output y is minimal

$$E = \frac{1}{N}\sum_{\mu=1}^{N} \log\left(\frac{1}{2}(1 + y_\mu o_\mu + \epsilon)\right)$$

Typical neural network circa 2005

ANN architecture: heuristic selection based on complexity adjustment and parameter estimation





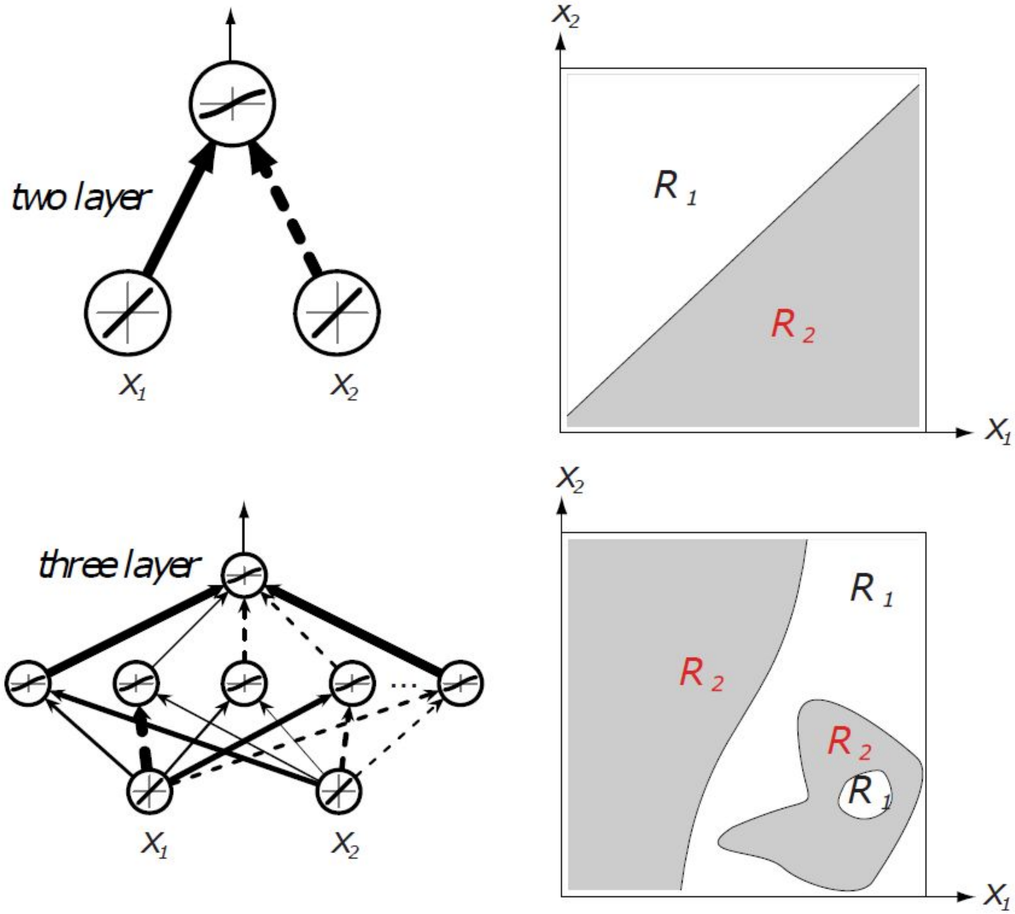
Theoretical basis:

Arnold - Kolmogorov (1957): if f is a multivariate continuous function, then f can be written as a finite composition of continuous functions of a single variable and the binary operation of addition

Gorban (1998): it is possible to obtain arbitrarily exact approx. of any continuous function of several variables using operations of summation and multiplication by number, superposition of functions, linear functions and one arbitrary continuous nonlinear function of one variable.

Typical neural network circa 2005

ANN architecture: heuristic selection based on complexity adjustment and parameter estimation



An example of a two and three-layer networks with two input nodes. Given an adequate number of hidden units, arbitrary nonlinear decision boundaries between regions R1 and R2 can be achieved

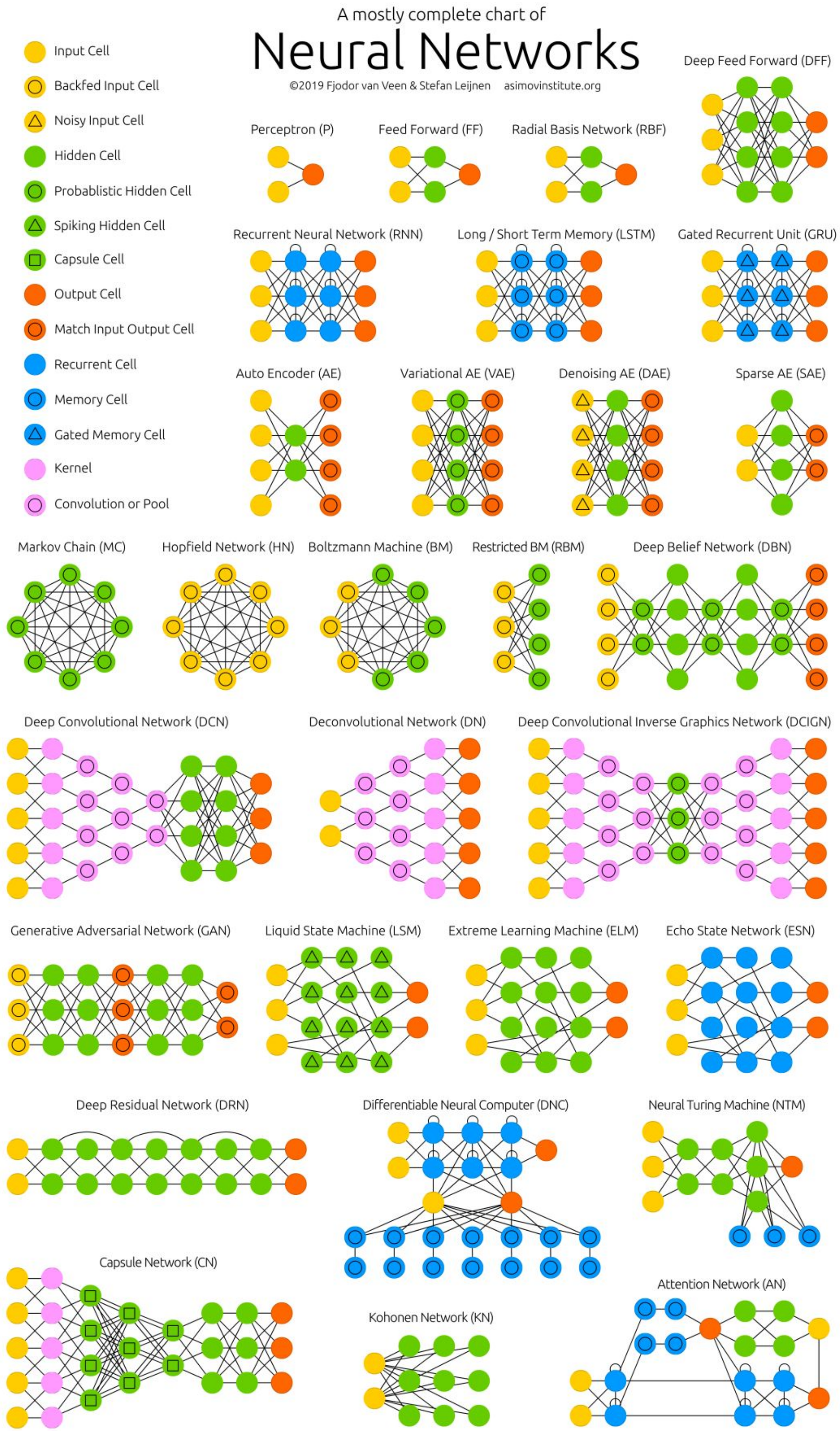
Theoretical basis:

Arnold - Kolmogorov (1957): if f is a multivariate continuous function, then f can be written as a finite composition of continuous functions of a single variable and the binary operation of addition

Gorban (1998): it is possible to obtain arbitrarily exact approx. of any continuous function of several variables using operations of summation and multiplication by number, superposition of functions, linear functions and one arbitrary continuous nonlinear function of one variable.

Neural Network is an universal approximator for any continuous function

Deep neural network circa 2020

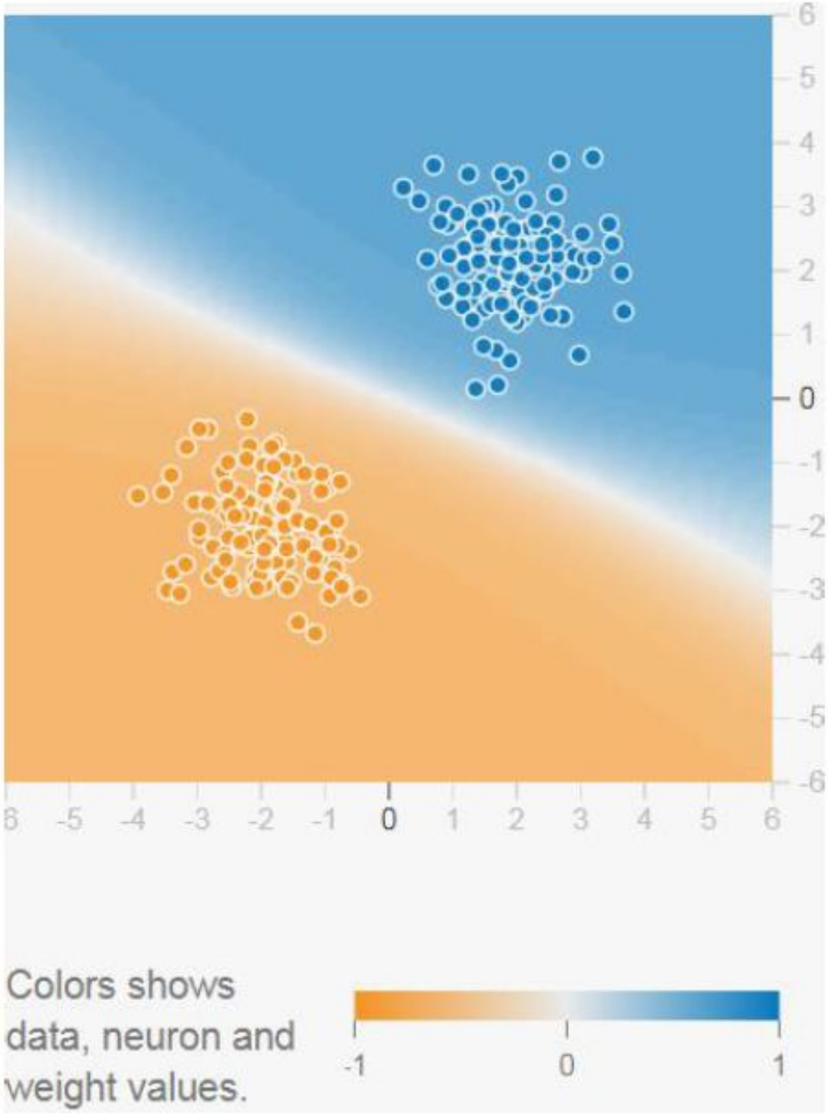


DNN architecture: Structure of the networks, and the node connectivity can be adapted for problem at hand

Convolutions: shared weights of neurons, but each neuron only takes subset of inputs

Difficult to train, only recently possible with large datasets, fast computing (GPU) and new training procedures / network structures

http://www.asimovinstitute.org/neural-network-zoo/



Orange shows negative values

Bue shows positive values

The data points (represented by small circles) are initially colored orange or blue, which correspond to positive one and negative one.

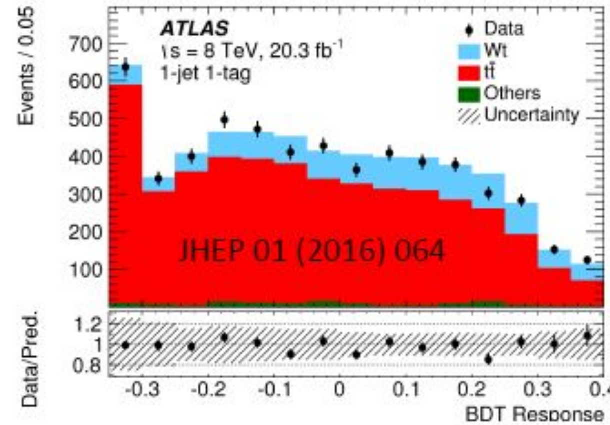
https://playground.tensorflow.org

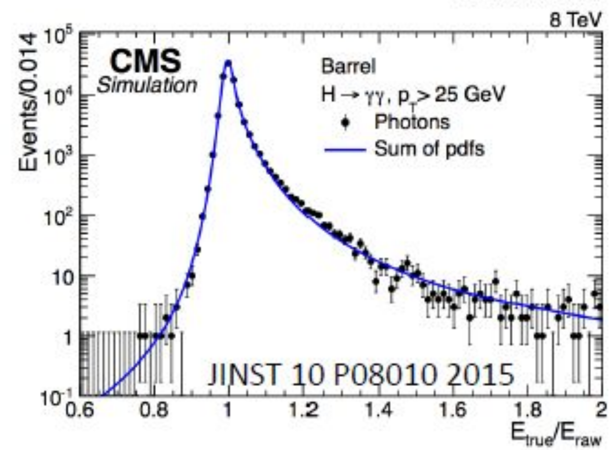
- **In analysis:**
  – Classifying signal from background, especially in complex final states
  – Reconstructing heavy particles and improving the energy / mass resolution

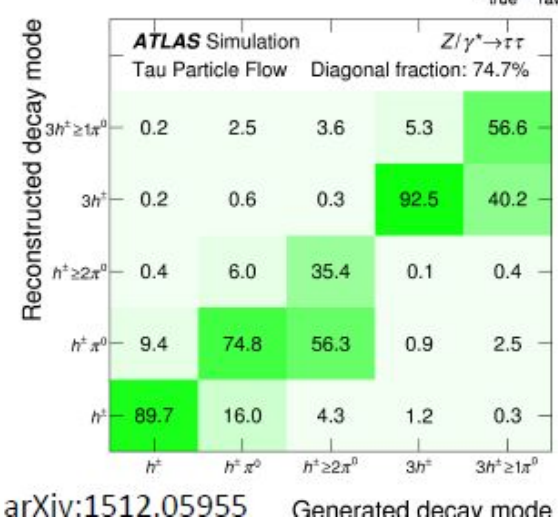
- **In reconstruction:**
  – Improving detector level inputs to reconstruction
  – Particle identification tasks
  – Energy / direction calibration

- **In the trigger:**
  – Quickly identifying complex final states

- **In computing:**
  – Estimating dataset popularity, and determining needed number and best location of dataset replicas

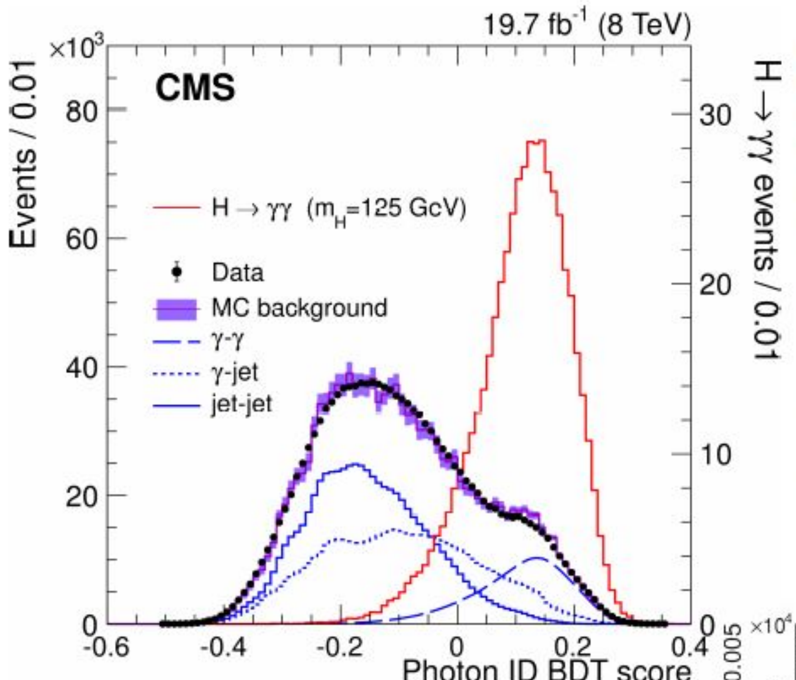






arXiv:1512.05955

BDT used for photon at CMS for ID (classification) and energy reconstruction (semi-parametric regression).



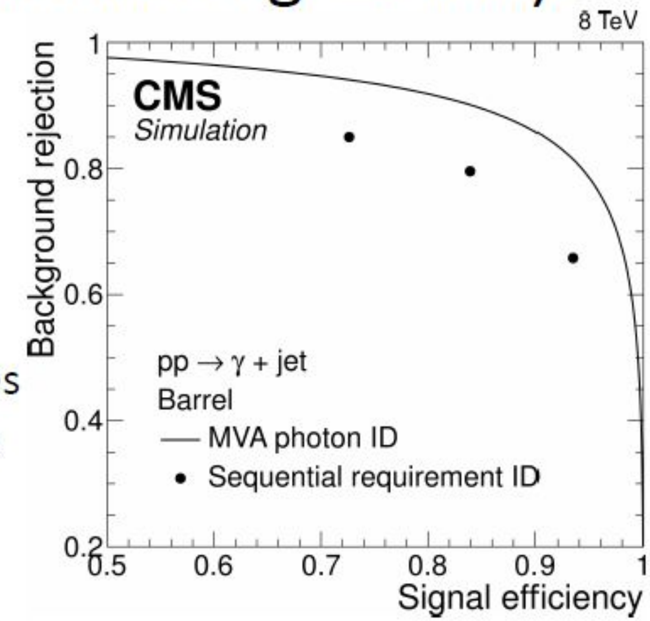
CMS Trained BDT in γ+jet MC

S=prompt γ

B=non-prompt γ

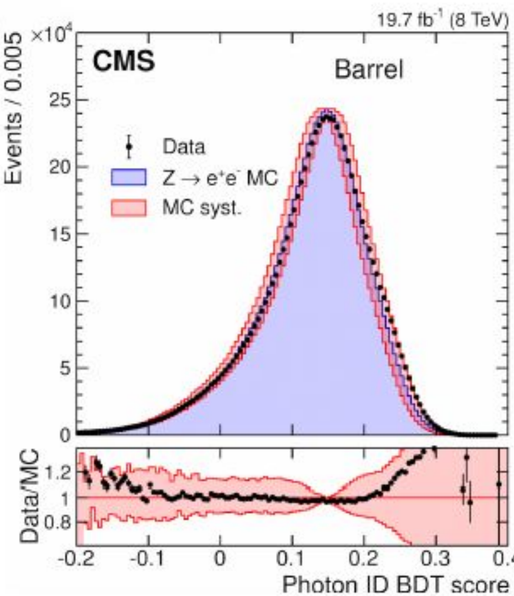
Inputs:

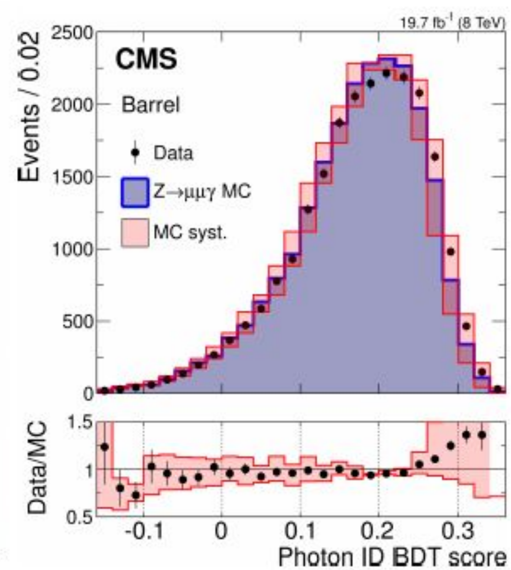
- all shower variables
- Isolation variables
- Eta, Et, etc



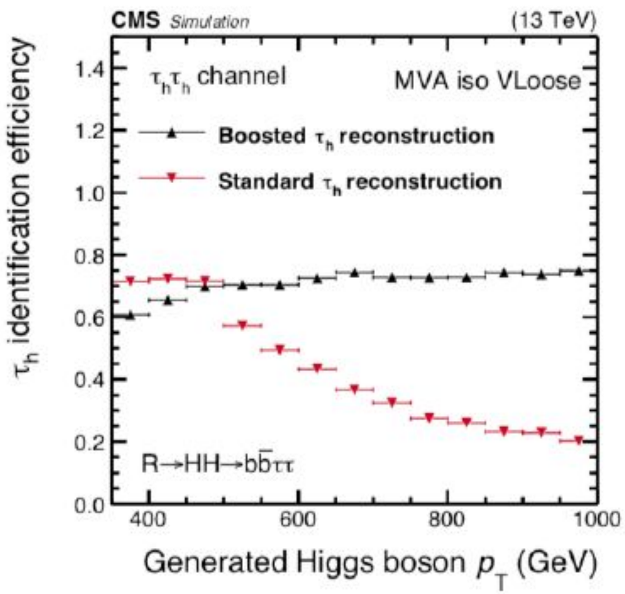
CMS also calibrates energy With a semi-parametric regression. [CMS-EGM-14-001]

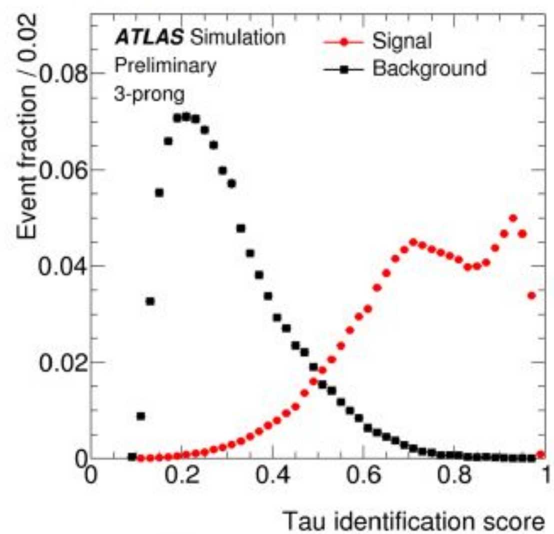
ATLAS uses BDT for photon calibration, same approach as for electron. [PERF-2013-05]. [ATL-PHYS-PUB-2017-022].

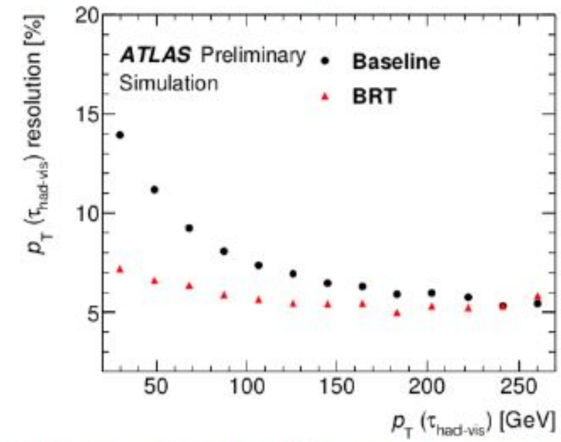




BDT for hadronic tau at CMS for ID (classification), at ATLAS for ID & energy calibration (regression).



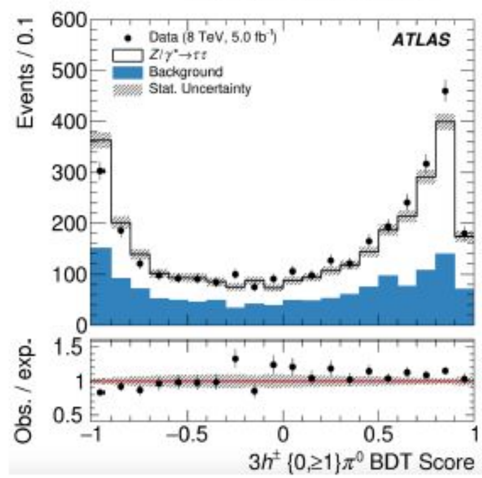




ATLAS BDT (BRT) regression improves resolution.

CMS & ATLAS each two BDTs for ID:

- tau (had) vs jet (q, gluon)
- tau (electron) vs electron
- Also boosted di-tau reco.
- CMS-TAU-16-003;
- ATL-PHYS-PUB-2015-045

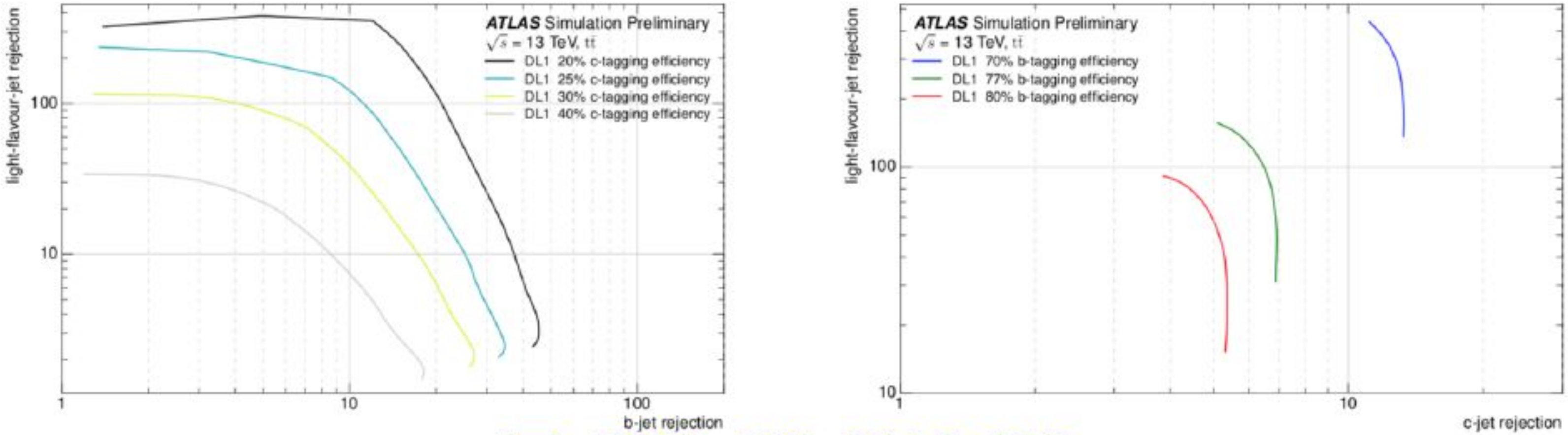


Inputs from baseline method, plus tau particle flow (using tracks for low pT), plus other calorimeter and tracking variables. ATLAS-CONF-2017-029

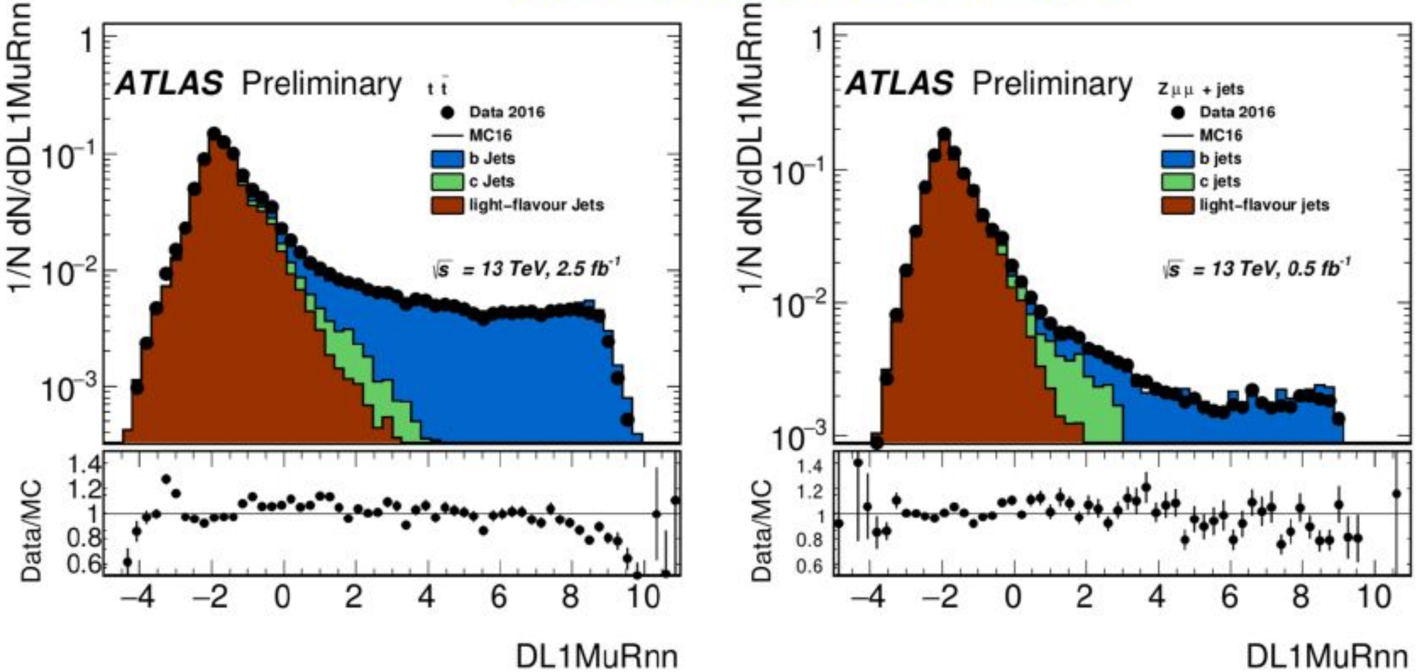
- ATLAS differentiate different decay modes of already identified tau by counting pi^0 PERF-2014-06.

Tau group was first in ATLAS to introduce a BDT ID at trigger level.

Deep neural network (DNN) vs BDT for b-tagging. For b-tagging similar performance, opens R&D.



[ATL_PHYS_PUB_2017_013]

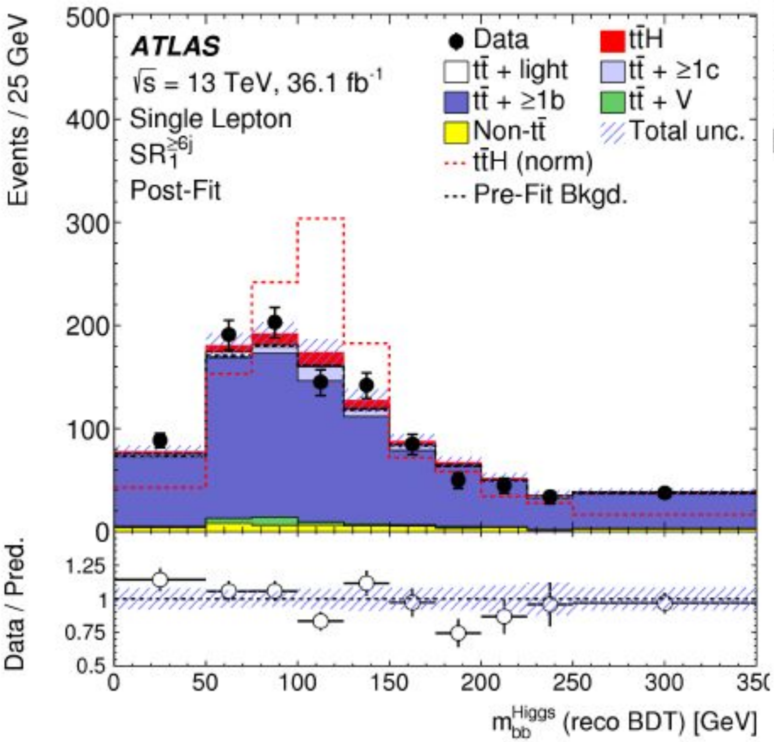


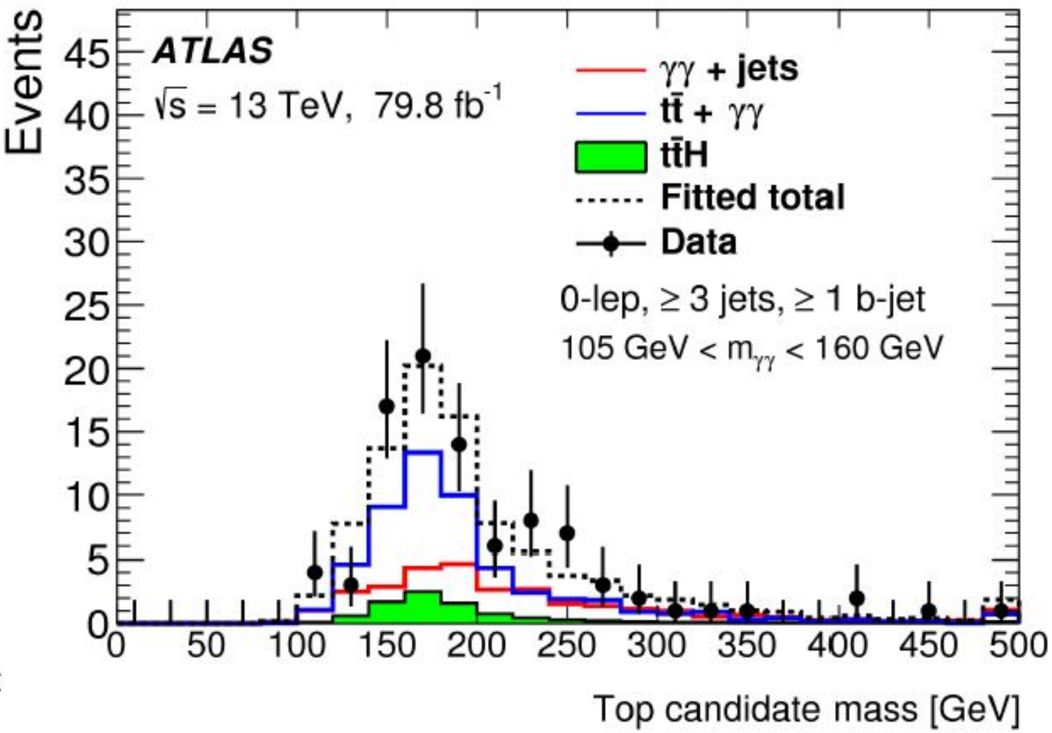
Same inputs, but at lower level.

Three outputs as probabilities of b, c, l (no tau).

Can be used for c-tagging too.

Example of BDT classification to choose which jets to group to reconstruct Higgs or top quark candidate mass.





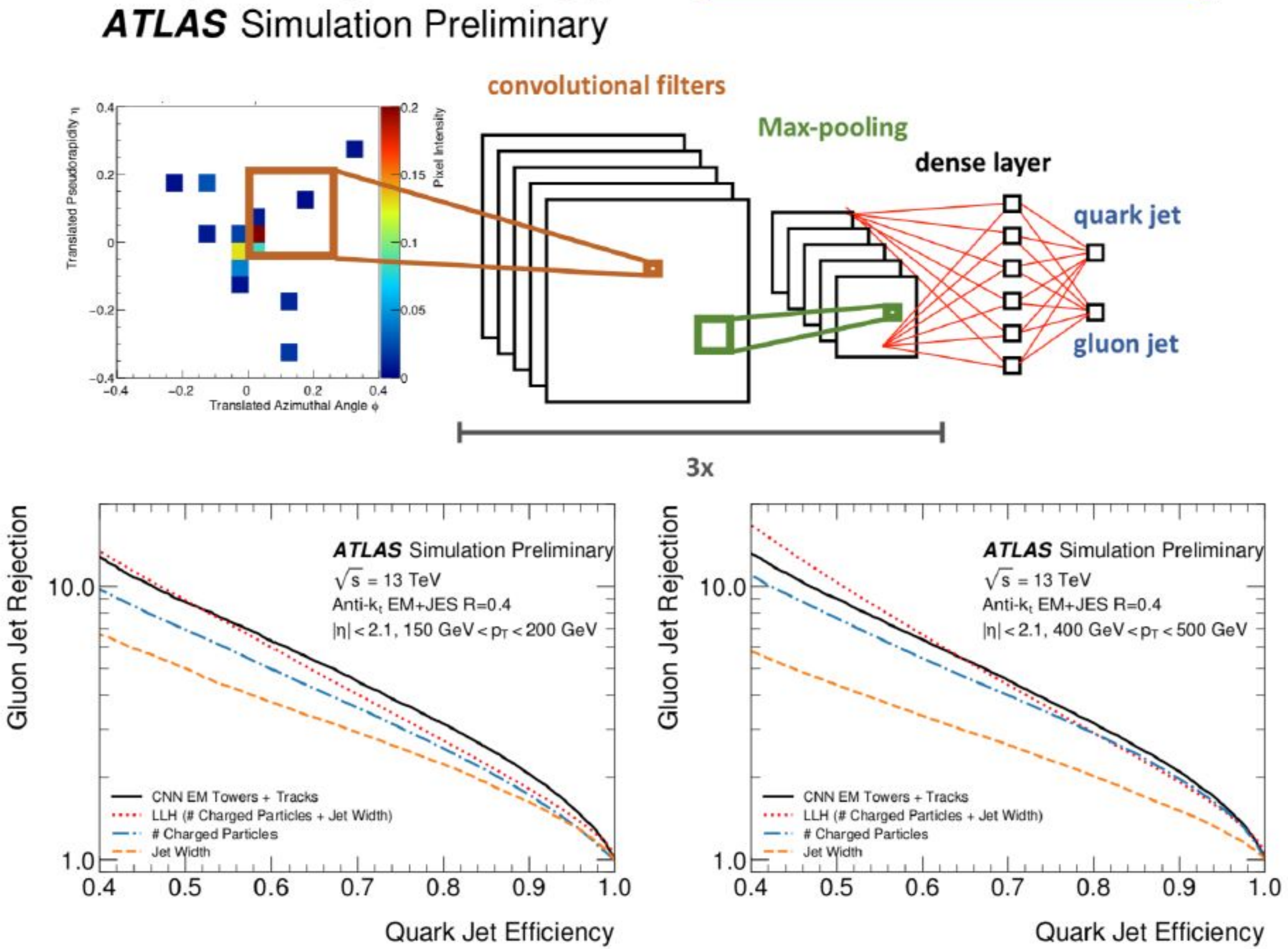
ATLAS ttH(bb) **H(bb) reconstruction**
Classification BDT choosing the right pairings of jets to form the H(bb) candidate in the resolved categories.

Right matching in about 30-50% of Higgs signal events. arXiv:1712.08895

ATLAS ttH(gamma gamma)
top quark mass reconstruction.
Classification BDT to choose which jets more likely to form the top quark candidate.

arXiv:1806.00425

Convolutional neural networks (CNN) classify jet images, like in the quark/gluon tagger (ATL-PHYS-PUB-2017-017).



Convolutional neural networks assume (translational) invariances as found in images. Images are scanned with (learned) filter matrices.

	TMVA	TensorFlow	Theano	Scikit Learn	R	Spark ML	VW	libFM	RGF	Torch
ROOT [**T, C**]	✓	through conversion into other formats, see Table 2								
CSV [**F**]		✓	✓	✓	✓	✓	×	×	×	✓
libSVM [**M**]							×	✓	×	
VW [**M**]							✓			
RGF [**M**]									✓	
NumPy [**R**]	[65]	✓	✓	✓	✓	✓	×	×	×	✓
Avro [**S, R**]					✓	✓				
Parquet [**S, C**]					✓	✓				
HDF5 [**S**]		×	×	×						✓
R df [**R**]					✓					

PyROOT	Python extension module that allows the user to interact with ROOT data/classes. [69]
root_numpy	The interface between ROOT and NumPy supported by the Scikit-HEP community. [65]
root_pandas	The interface between ROOT and Pandas dataframes supported by the DIANA/HEP project. [70]
uproot	A high throughput I/O interface between ROOT and NumPy. [71]
c2numpy	Pure C-based code to convert ROOT data into Numpy arrays which can be used in C/C++ frameworks. [72]
root4j	The hep.io.root package contains a simple Java interface for reading ROOT files. This tool has been developed based on freehep-rootio. [73]
root2npy	The `go-hep` package contains a reading ROOT files. This tool has been developed based on freehep-rootio. [73]
root2hdf5	Converts ROOT files containing TTrees into HDF5 files containing HDF5 tables. [74]

https://arxiv.org/pdf/1807.02876.pdf

References

https://arxiv.org/pdf/1807.02876.pdf

http://www-group.slac.stanford.edu/sluo/Lectures/Stat2006_Lectures.html

https://indico.cern.ch/event/77830/

http://www.pp.rhul.ac.uk/~cowan/stat/cowan_weizmann10.pdf

https://web.stanford.edu/~hastie/ElemStatLearn/

https://cds.cern.ch/record/2651122

http://cds.cern.ch/record/2634678

http://cds.cern.ch/record/2267879/