Machine Learning technologies for INFN

Lucio Anderlini

Consiglio di Sezione
2020-02-07

Istituto Nazionale di Fisica Nucleare
SEZIONE DI FIRENZE

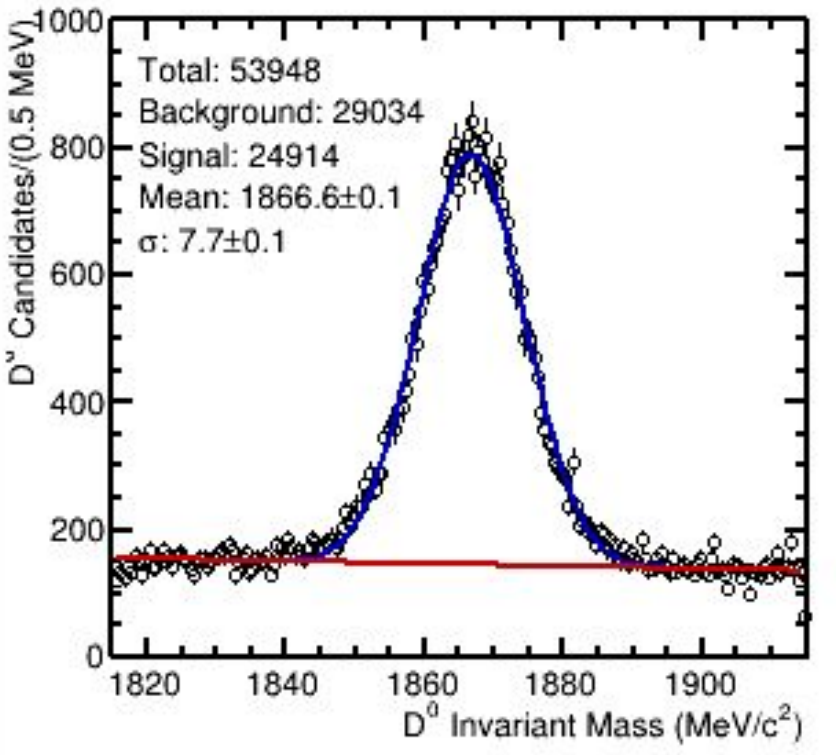
Unbinned Maximum Likelihood Fits as ML algorithms

The maximum likelihood principle is widely used in physics to model datasets.

The likelihood is defined as

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} f(x_i; \boldsymbol{\theta})$$

where the x_i represent the dataset, and $\boldsymbol{\theta}$ is a vector of **parameters** the probability density function f depends on.

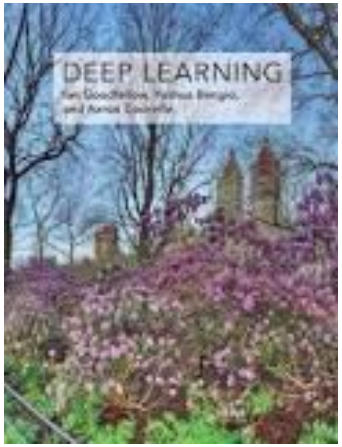


If you are interested in the values on $\boldsymbol{\theta}$ then it's **parametric density estimation**.

If $\boldsymbol{\theta}$ is just needed to define a shape, then it's **non-parametric density estimation**, a widely investigated research topic in machine learning.

* Picture from the LHC Masterclasses programme, for the LHCb experiment

Convergence. Machine Learning ***is*** fitting



"Most modern neural networks are trained using maximum likelihood. This means that the cost function is simply the negative log-likelihood."

I. Goodfellow, Y. Bengio and A. Courville, *"Deep Learning"*, MIT Press (2016)

zfit provides model building and fitting on top of TensorFlow.

www.tensorflow.org ▾ Traduci questa pagina

TensorFlow

An end-to-end open source machine learning platform.

zfit: scalable pythonic fitting

Jonas Eschle, Albert Puig Navarro, Rafael Silva Coutinho, Nicola Serra

Physik-Institut, Universität Zürich, Zürich (Switzerland)

https://arxiv.org/abs/1910.13429

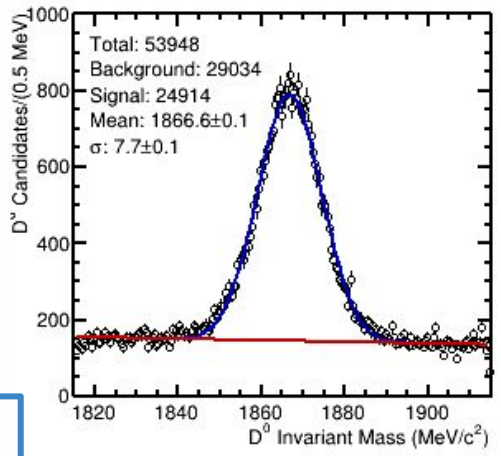
Abstract

Statistical modeling is a key element for High-Energy Physics (HEP) analysis. The standard framework to perform this task is the C++ ROOT/RooFit toolkit; with Python bindings that are only loosely integrated into the scientific Python ecosystem. In this paper, zfit, a new alternative to RooFit written in pure Python, is presented. Most of all, zfit provides a well defined high level API and workflow for advanced model building and fitting together with an implementation on top of TensorFlow. It is designed to be extendable in a very simple fashion, allowing the usage of cutting-edge developments from the scientific Python ecosystem in a transparent way. Moreover, the main features of zfit are introduced, and its extension to data analysis, especially in the context of HEP experiments, is discussed.

From Fitting to Artificial Intelligence: *optimization algs*

Modern "artificially intelligent" stuff requires fit functions with millions to billions of free parameters.

As for fitting, the functions are optimized with *gradient descent*, but exploiting some tricks.



Stochastic Gradient Descent

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} f(x_i; \boldsymbol{\theta})$$

At each evaluation of the likelihood, only a subset of *n* entries is used.

The computational cost scales with *n*
The error on the gradients scales with $\sqrt{n}$

?

Momentum based optimization

*The trajectory explored in the optimization process is computed assuming a "*mass*" giving inertial properties.*
Statistical uncertainties on the gradients cancel in the process.

From Fitting to Artificial Intelligence: *fit function*

Non-parametric functions are chosen *very* carefully to allow fast and **analytical computation** of the gradients $\nabla_\theta L(\theta)$

Weight matrix

Activation function

Input features

Another weight vector

Bias vector

$$f(\mathbf{x}; \mathbb{A}, \mathbf{b}) = w^T \sigma\left(\mathbb{A}\mathbf{x} + \mathbf{b}\right)$$

The activation function σ must be non linear and simple to differentiate. A typical choice is the **Rectified Linear Unit** function

$$\sigma(z) = \max(0, z)$$

From a computational point of view, very large matrices are difficult to treat. It is preferred to split the problem with function composition

$$f(\mathbf{x}; w^{\mathrm{T}}, \mathbb{A}_1, \mathbf{b}_1, \mathbb{A}_d, \mathbf{b}_d, \ldots, \mathbb{A}_d, \mathbf{b}_d) = w^T \sigma\left(\mathbb{A}_{\mathrm{d}} \sigma\left(\ldots\left(\mathbb{A}_2 \sigma\left(\mathbb{A}_1 \mathbf{x} + \mathbf{b}_1\right) + \mathbf{b}_2\right)\ldots\right) + \mathbf{b}_d\right)$$

This function is a deep feed-forward neural network of depth *d*

Specializing the Neural Network to the task (CNNs)

Sometimes, special matrices can help describing the problem more effectively.

In particular using “filters” with learned weights is extremely effective.

DFT can be used to compute effectively the effect of these filters as discrete **convolutions**.

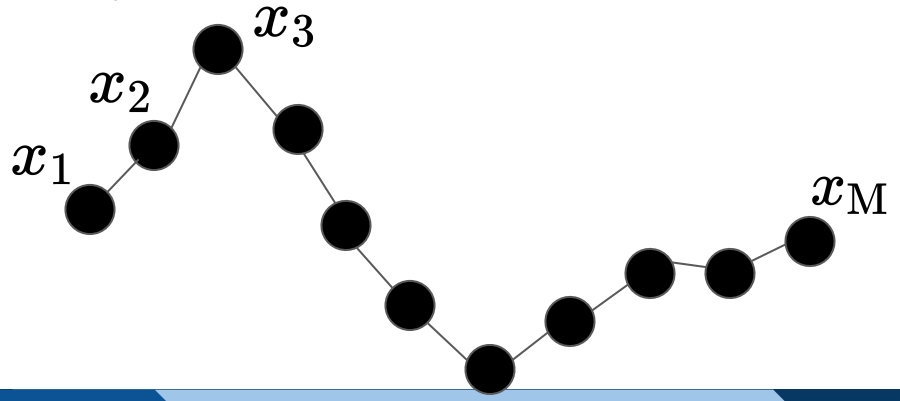
The simplest filter is a 3-weight filter that can be trained for example to distinguish local maxima from local minima.

$$f(\mathbf{x}; \mathbb{A}, \mathbf{b}) = w^T \sigma\left(\mathbb{A}\mathbf{x} + \cancel{\mathbf{b}}\right)$$

$$\mathbb{A}\mathbf{x} = \begin{pmatrix} a_0 & a_1 & 0 & 0 & \cdots & 0 \\ a_{-1} & a_0 & a_1 & 0 & \cdots & 0 \\ 0 & a_{-1} & a_0 & a_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{\mathrm{M}} \end{pmatrix}$$

Maxima “detector” with:

$$\begin{cases} a_{-1} = -1 \\ a_0 = 2 \\ a_1 = -1 \end{cases} \Rightarrow y_i = (x_i - x_{i-1}) + (x_i - x_{i+1})$$

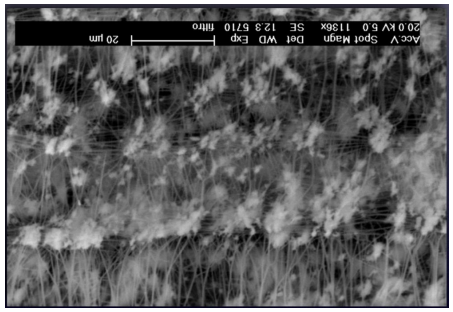


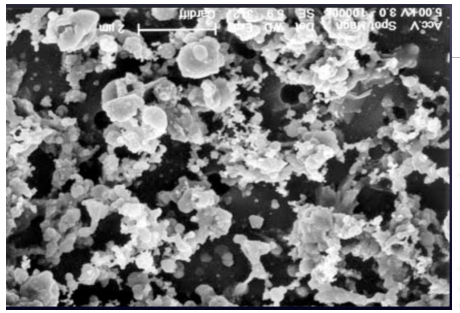
The ability to learn to identify complex features in the data exploiting the concept of neighbourhood is known as **representation learning**.

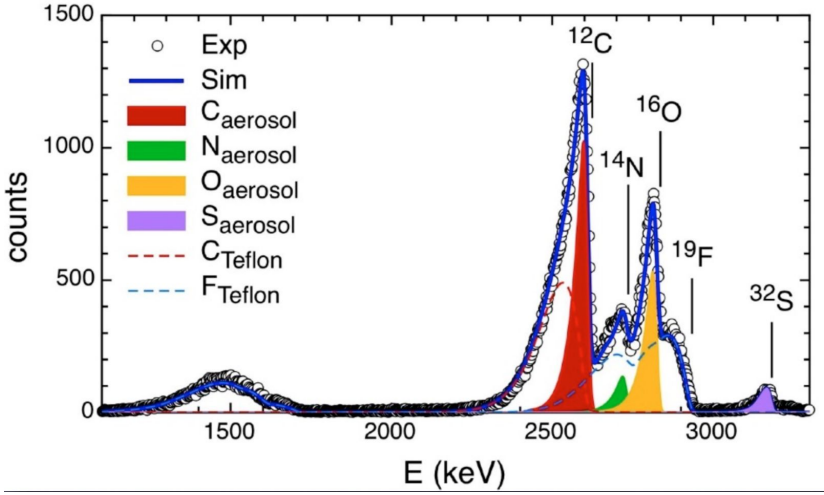
Ion Beam Analysis (LABEC)

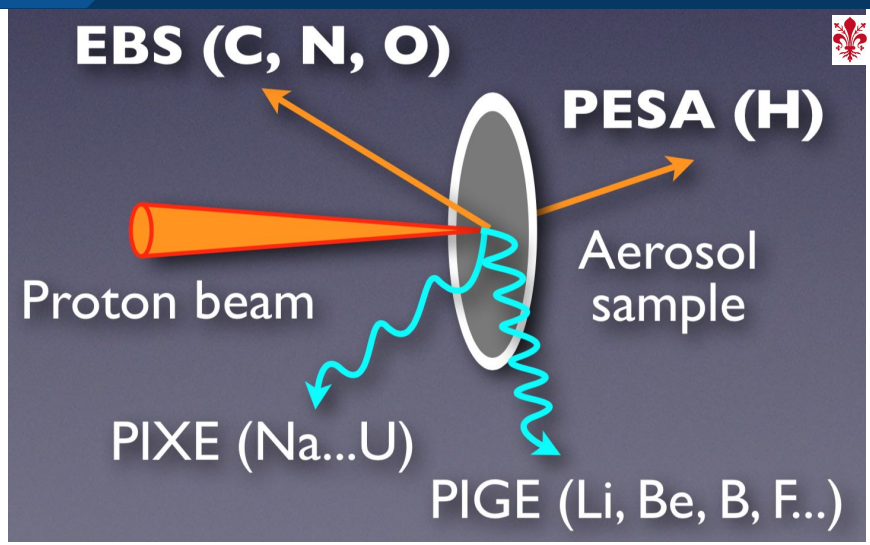
Convolutional Neural Networks are being explored for the automatization of the spectral analysis of the Elastic Backscattering Spectrometry (EBS) spectra of aerosol on teflon filters.

SEM images of a teflon filter



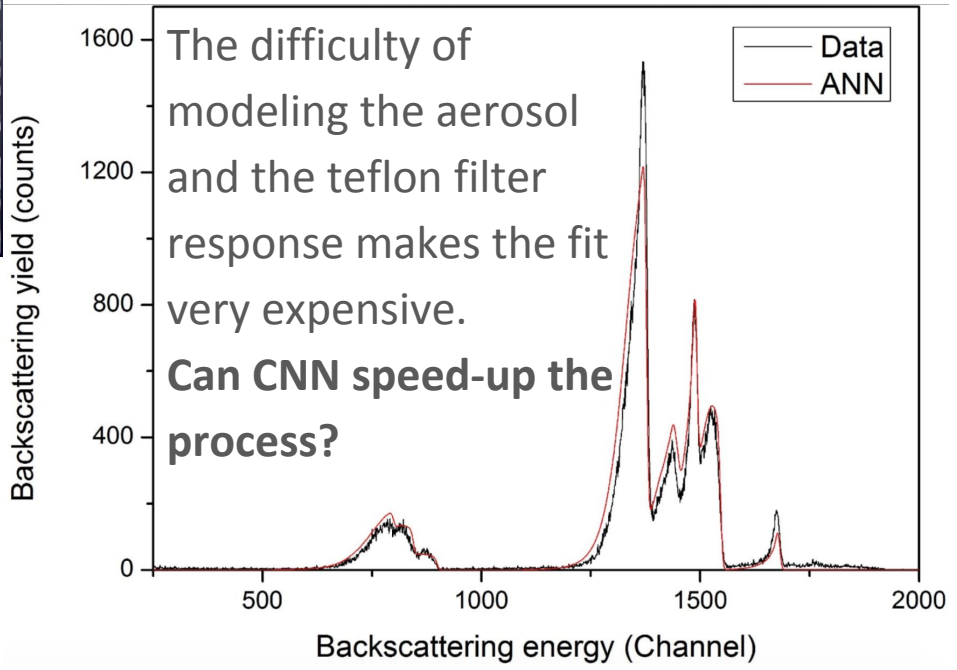






The difficulty of modeling the aerosol and the teflon filter response makes the fit very expensive.

Can CNN speed-up the process?



Successful proof of principle with an available training set. Work ongoing for a **better simulation**.

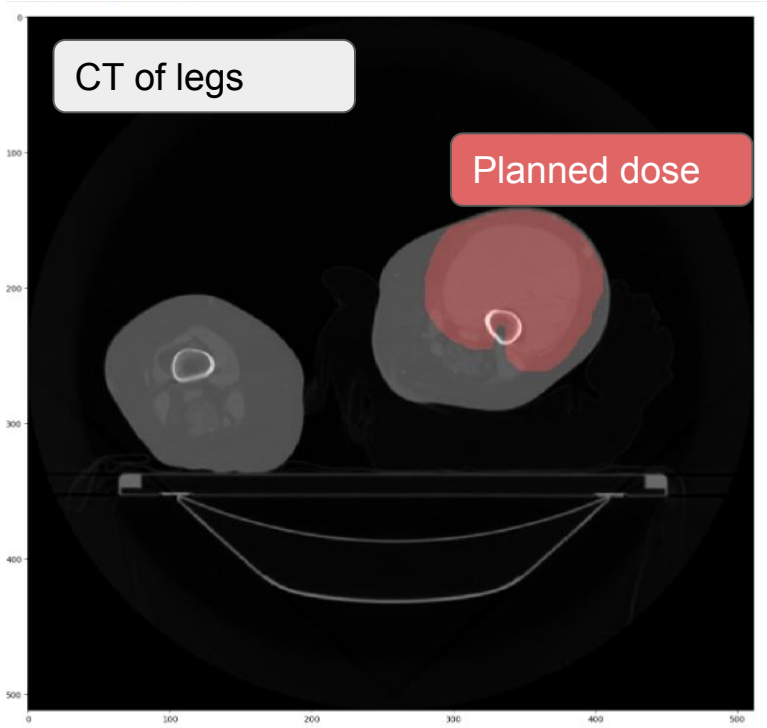
Image Processing

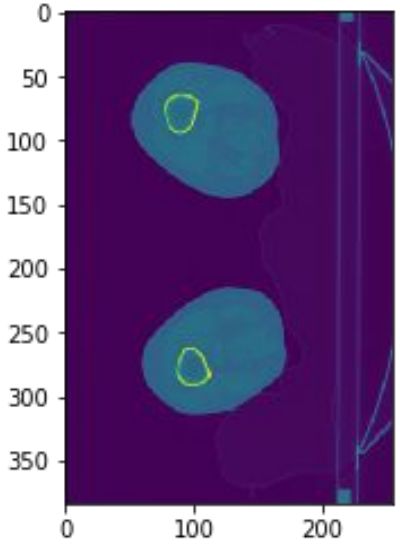
The approach can be extended to identify structures in three dimensions, to process with CNNs the result of CT, NMR and PET trying to predict the outcome of radiotherapy on healthy tissues.

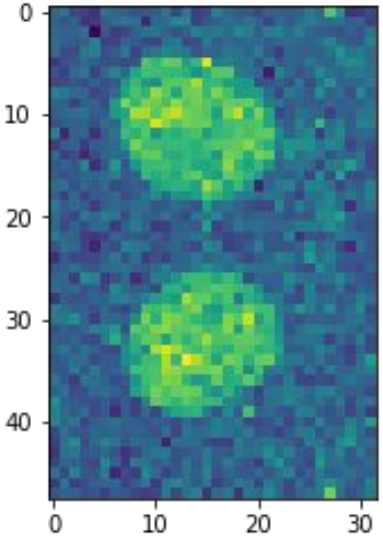
Many works in the literature processing 2D slices which is an effective but incomplete approach.

Processing 3D objects is extremely challenging for the large RAM required. With a 100 GB server and one hour of processing, we are able to train a network that barely identifies legs.

Developing less memory greed algorithms is a major challenge that INFN is undertaking.







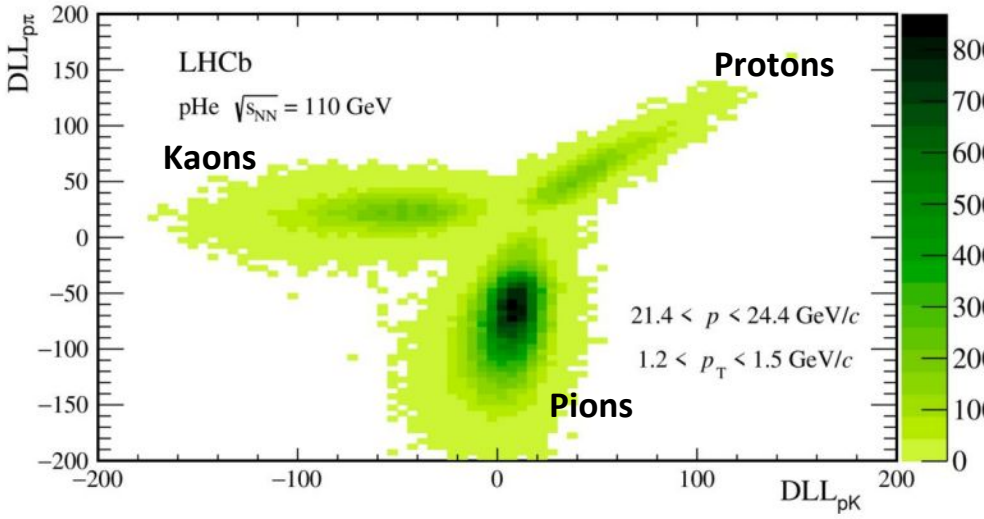
Infer the *probability density function*

Neural Networks in Density Estimation

Neural networks are not normalized, but nothing prevents from combining analytically normalized *pdf*s with neural networks.

Problem: model the response of the RICH detectors of LHCb.

The response depends on the track momentum *p,* pseudorapidity η and detector occupancy *nTracks*.

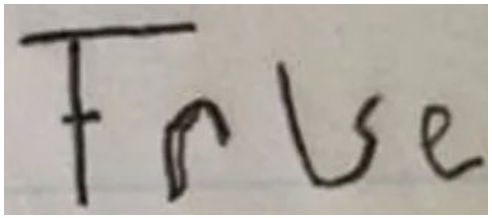


w, μ_K μ_π V_K $V_{K\pi}$ and V_π are **non-parametric functions** of *p*, η and *nTracks,* modeled as neural networks.

$$f(\mathrm{DLL}_{pK}, \mathrm{DLL}_{p\pi}) = \sum_i w_i G\left((\mathrm{DLL}_{pK} - \mu_{K,i} \quad \mathrm{DLL}_{p\pi} - \mu_{\pi,i}) \begin{pmatrix} V_{K,i} & V_{K\pi,i} \\ V_{K\pi,i} & V_{\pi,i} \end{pmatrix}^{-1} \begin{pmatrix} \mathrm{DLL}_{pK} - \mu_{K,i} \\ \mathrm{DLL}_{p\pi} - \mu_{\pi,i} \end{pmatrix} \right)$$

The resulting *pdf* has **O(10^5) free parameters** fitted using TensorFlow on calibration data to construct **kinematic-dependent templates**, used to count protons in unfiltered samples

From Density Estimation to Classification



Often, neural networks are used as classifiers.

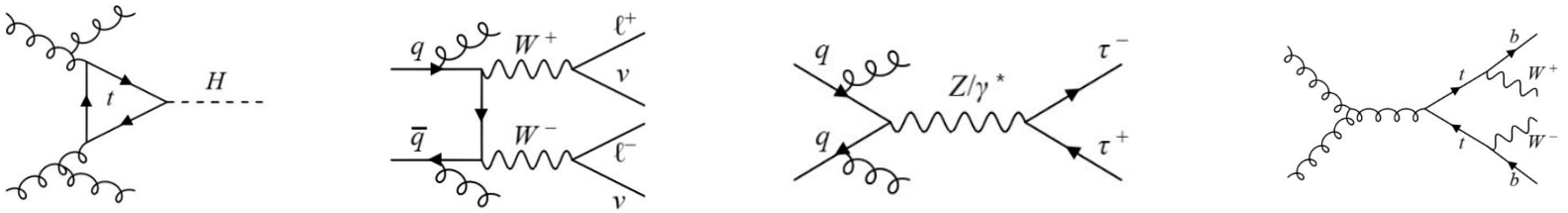
The *probability mass function* is the Bernoulli distribution, with a probability depending on some parameters *x*. Dependence is modeled with a neural network.

$$pmf(b,x) = \begin{cases} p(x) & \text{if } b = 1 \\ 1 - p(x) & \text{if } b = 0 \end{cases}$$

$$-\log \mathcal{L} = \sum_{i \text{ with } b=1} -\log[p(x_i)] + \sum_{i \text{ with } b=0} -\log[1 - p(x_i)]$$

(also known as *Bernoulli Cross-Entropy*)

Extending to a multinomial distribution, CMS adopted neural network classifiers to classify *dijet events* in four categories with four different production mechanisms:



The ***x*** variables (27) describe the energy and the angles of reconstructed particle jets, of the leptons in the final state and invariant masses of the combinations.

The structure of the neural network (depth and rank of the matrices) was also optimized in a iterative optimization procedure.

Combining Neural Networks into Autoencoders for DQM

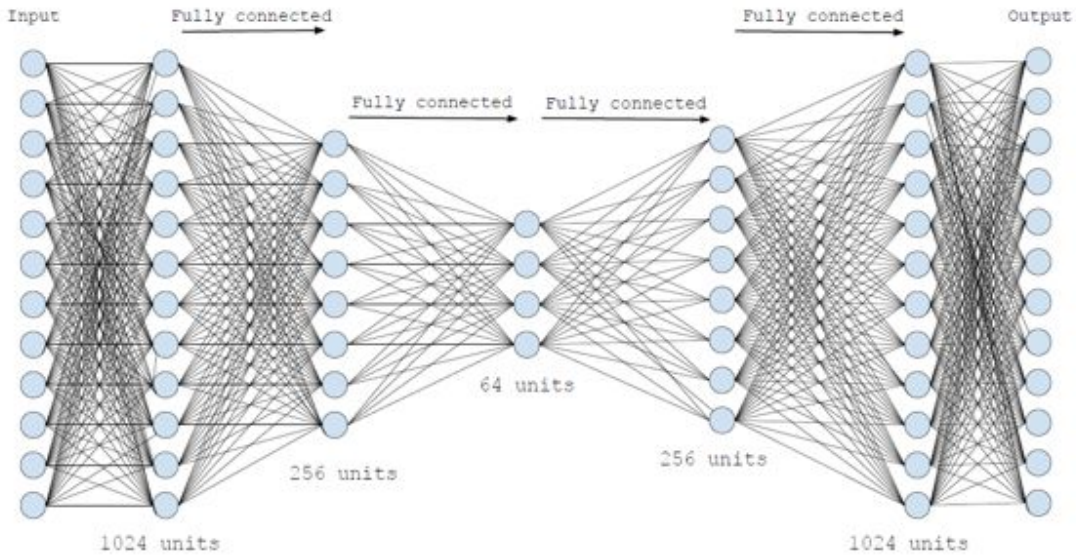
Sometime defining a probability and thus a likelihood is simply impossible, or too CPU expensive.

Autoencoders are neural networks trained to minimize the squared difference between input and output.

Undercomplete autoencoders encode shared features of the training sample in the weights of the network.

Hence, the trained neural network only works properly of test samples sharing the encoded features with the training sample.

It behaves as an **anomaly detector.**



CMS is using **anomaly detection** algorithms based on **autoencoders** to promptly detect anomalies in the sub-detector performance measurements.

Autoencoders are trained on human-classified data-taking runs.

If evaluated on unclassified runs the autoencoder results into an **output inconsistent with the input**, the run is flagged as anomalous.

Combining Neural Networks into GANs

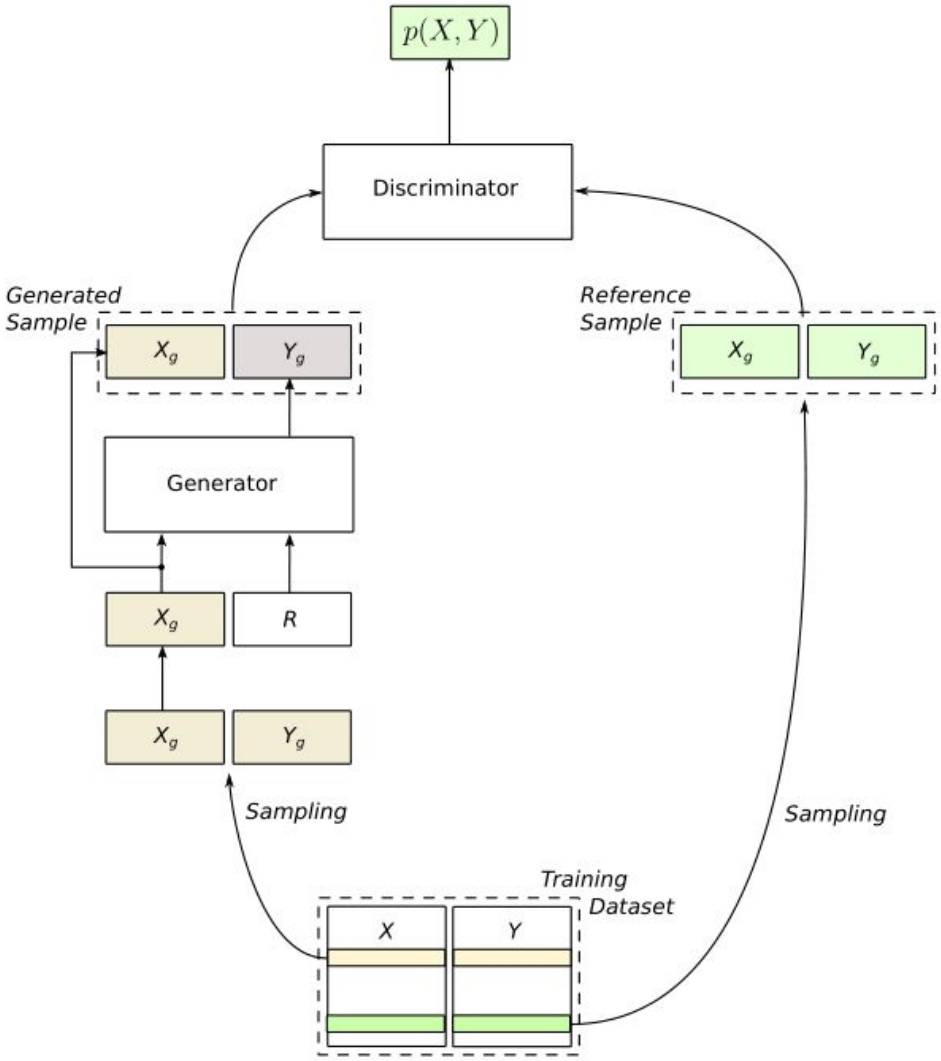
When writing the loss function bets too difficult, an alternative is to use a second **neural network to approximate the loss** function of the former neural network.

Generative Adversarial Networks (GANs) are the most famous example.

We want the neural network G to output **random entries** in a multidimensional parameter space, **with the same underlying *pdf* as the training sample.**

A second neural network *D* is trained as a classifier to distinguish generated and training samples.

G is trained to worsen the performance of D



Combining Neural Networks into GANs

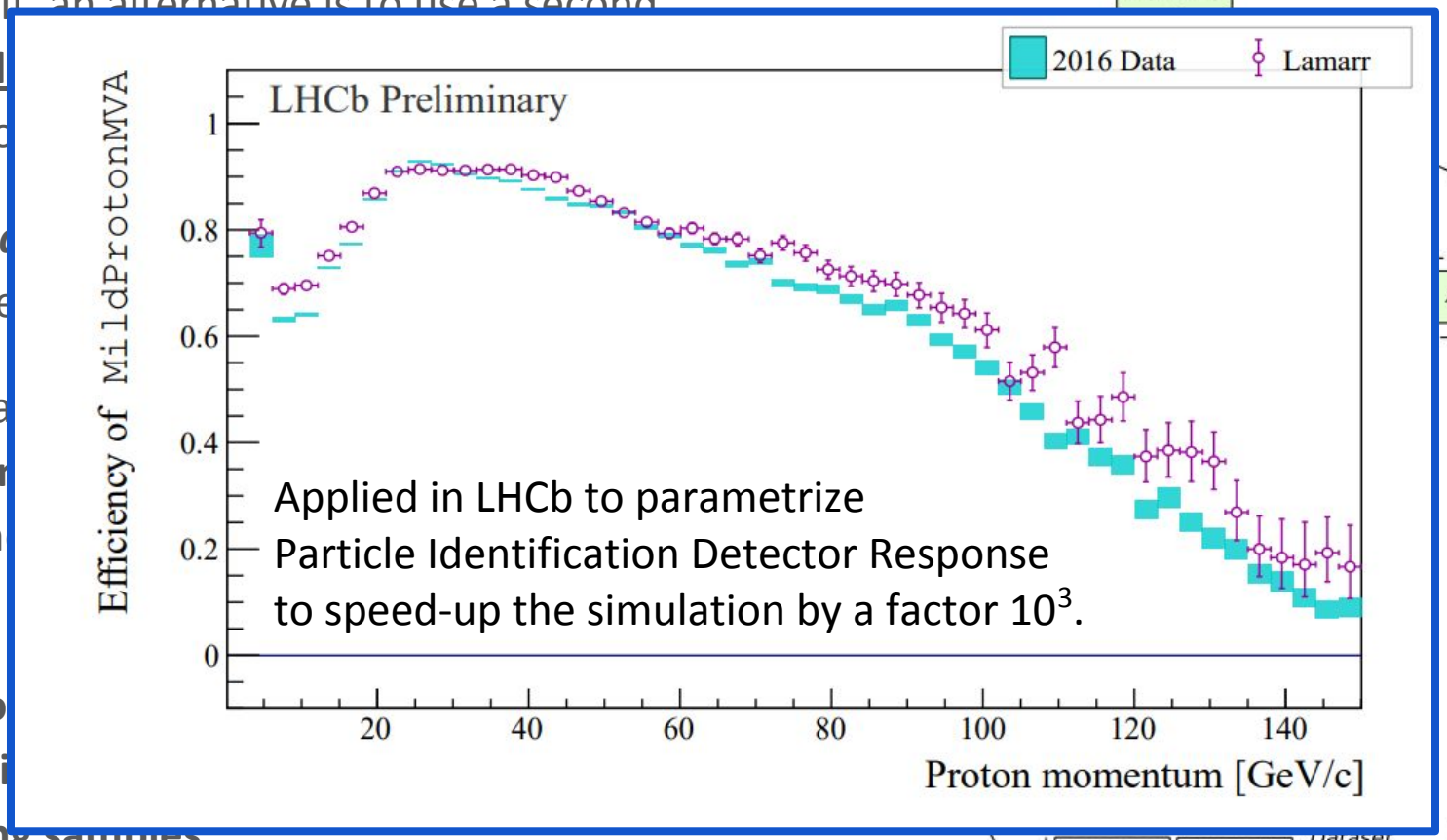
When writing the loss function bets too difficult, an alternative is to use a second **neural** functio

Genera are the

We wa **randor** param ***pdf*** as

A seco classifi training samples.

G is trained to worsen the performance of D

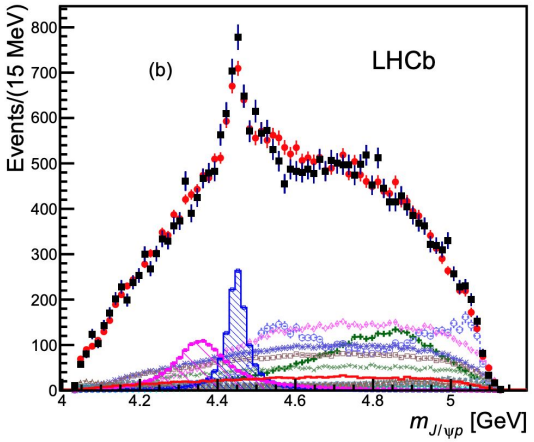


Beyond Neural Networks

ML algorithms without ML

QCD phenomenology

Observation of a pentaquark resonance: [PRL 115 (2015) 072001]

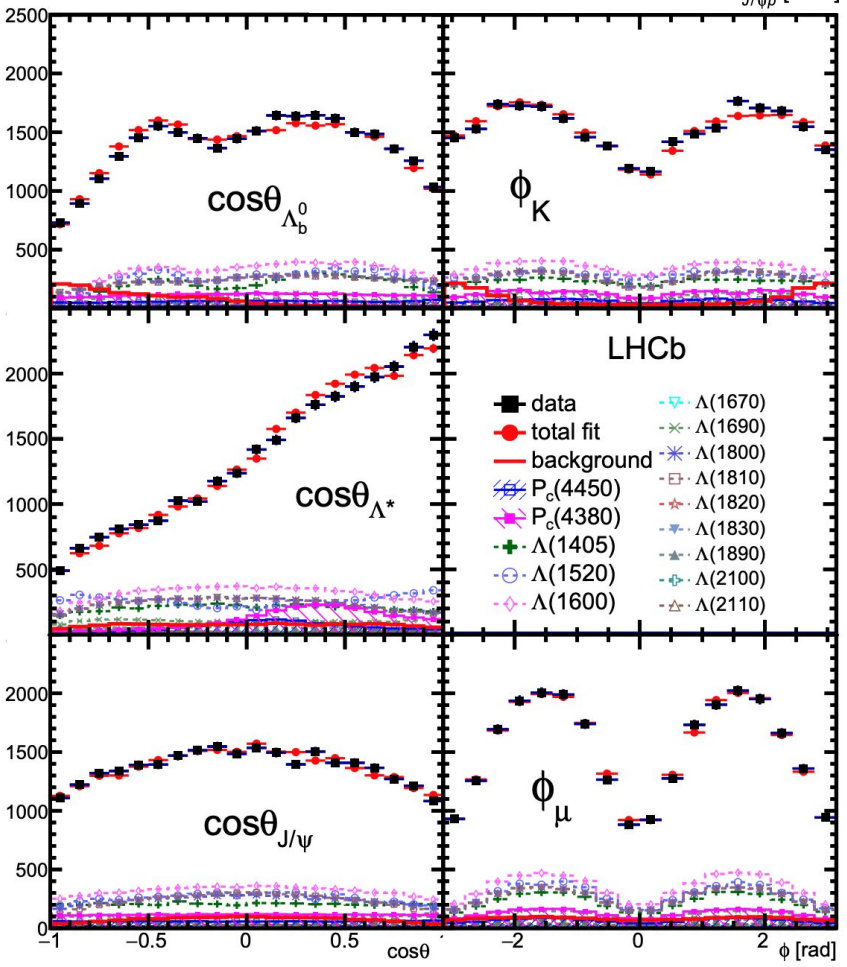


Search of resonances in the decay of heavy hadrons can exploit the complete amplitude analysis of the decay.

$$d\Gamma = \frac{1}{(2\pi)^5}\frac{1}{16M^2}\,|\mathscr{M}|^2\,|\boldsymbol{p}_1^*|\,|\boldsymbol{p}_3|\,dm_{12}\,d\Omega_1^*\,d\Omega_3$$

$$\mathscr{M} = \sum_i \alpha_i M_i(m_{12}, \Omega_1^*, \Omega_3)$$

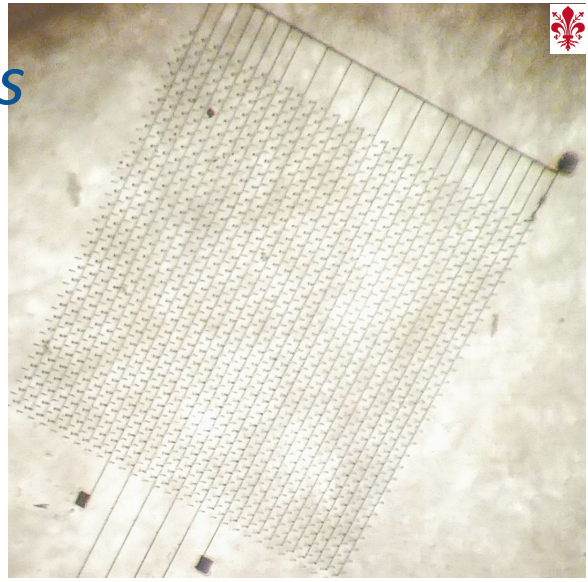
To interpret the experimental dataset the model is fitted to the data with free α_i and, where not known, mass and width of the resonances.



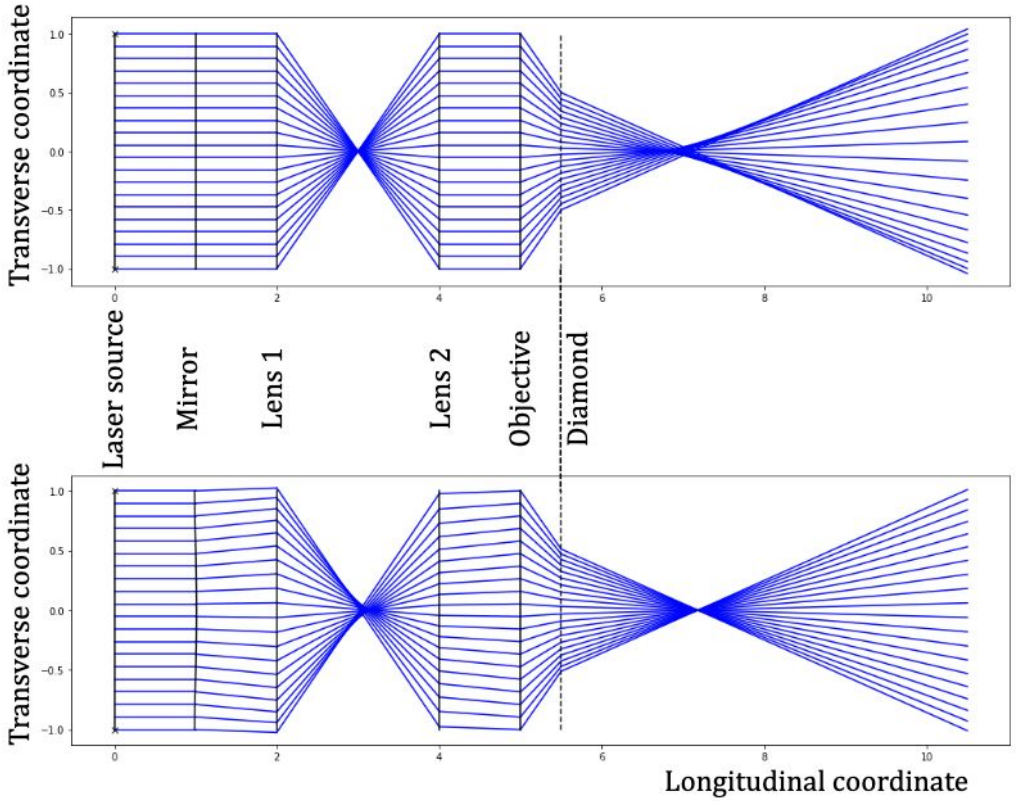
TensorFlow was interfaced to **MINUIT** to combine the powerful computational part with the **well established procedure** for statistical error determination determination.

MINUIT: 30 h ⟶ MINUT + TF: 2 h

Beyond Neural Networks, *adaptive optics*



To produce time-resolved diamond sensor, a laser is focused within a diamond sample to *burn* the diamond into graphite. A correction of the optical aberration introduced by the diamond is needed to achieve decent resistivity of columns.



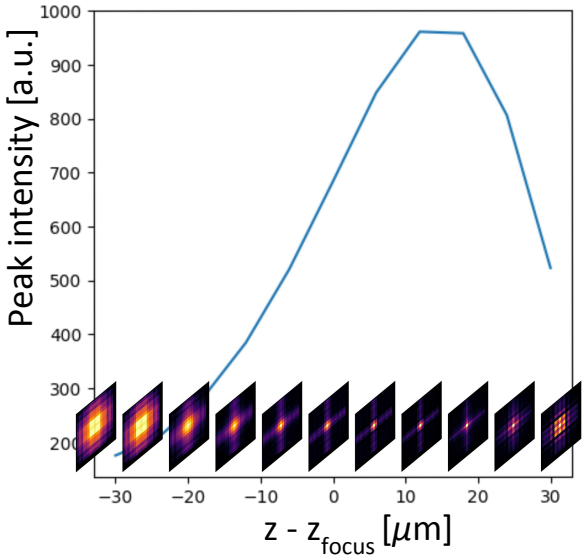
In the ongoing upgrade, we will use a Space Light Modulator (SLM) as adaptive optics element.
SLM defines the shift of the laser wave on a 800x600 pixel matrix. This allow to create complex patterns on the focal plane.

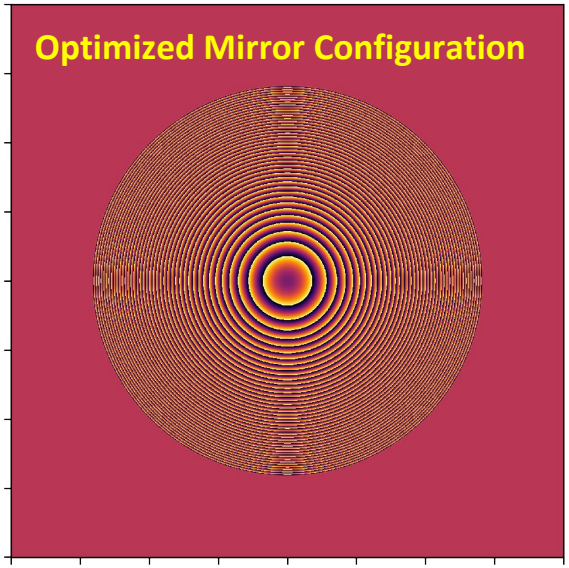
Solution in 3D requires assumptions that we would like to avoid.

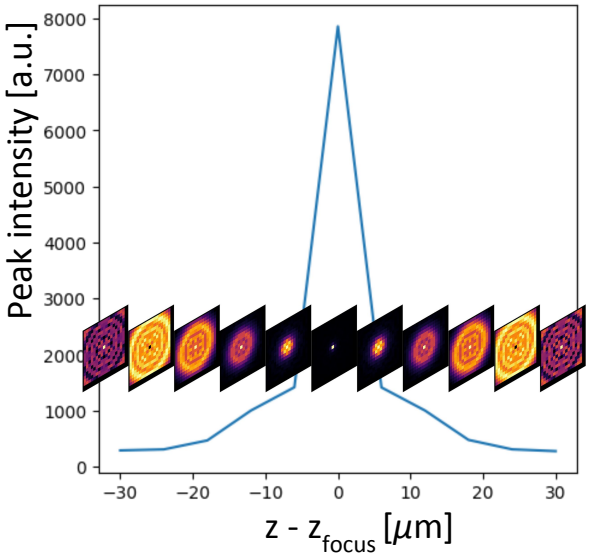
Beyond Neural Networks, *adaptive optics*

The 800x600 “picture” defining a space-dependent phase shift has been included in a **TensorFlow-based simulation as a matrix**, as a set of weights of a Neural Network layer.

The 3D energy distribution obtained from the simulation is then compared to a target in an iterative optimization procedure based on the same optimizer as those used to **train neural networks.**







Successful proof-of-principle, but we coded the simulation with too crude approximations for the result to be usable on the real system.

Technological developments

ML_INFN and INFN/Cloud: *federation and virtualization*

Today we share data, code snippets, tutorials using a dismissed server bought for web services.

It is accessible from anyone from within the INFN network: http://acer-1.fi.infn.it:8000

The increasing need for more resources, including GPU and fast access to large datasets, is being addressed **at national level with ML_INFN** (CSN5) and **INFN/Cloud** (CCR).

High-Performance and High-Throughput Computing resources available throughout INFN might be federated network with cloud technology relying on **virtualization**.

INFN is developing a **backbone of computing resources** hosted at CNAF and in Bari that can be connected to a number of lesser computing resources, including provisional resources from **private companies** to mitigate peaks in the request.

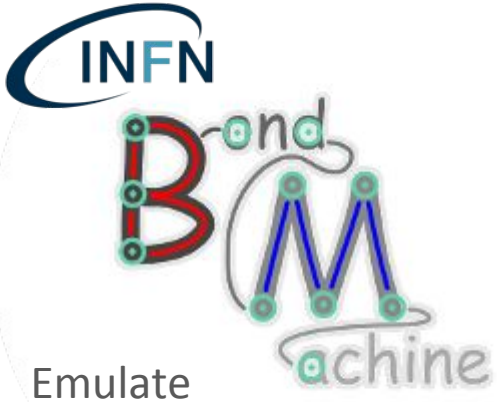
The INFN/Cloud has developed a **technological roadmap** and has identified the legal **critical points** due to the **sharing of data**. Sometime including private **medical data**.

Deployment on FPGAs for low power inference

Powerful computing resources are needed for training, but trained networks can be deployed on programmable hardware to save power.

This is very promising for the real time processing of events at large experiments, where the computing power is becoming a non-negligible cost.

Three different approaches :



Emulate mini-processors on the FPGA to read binary code

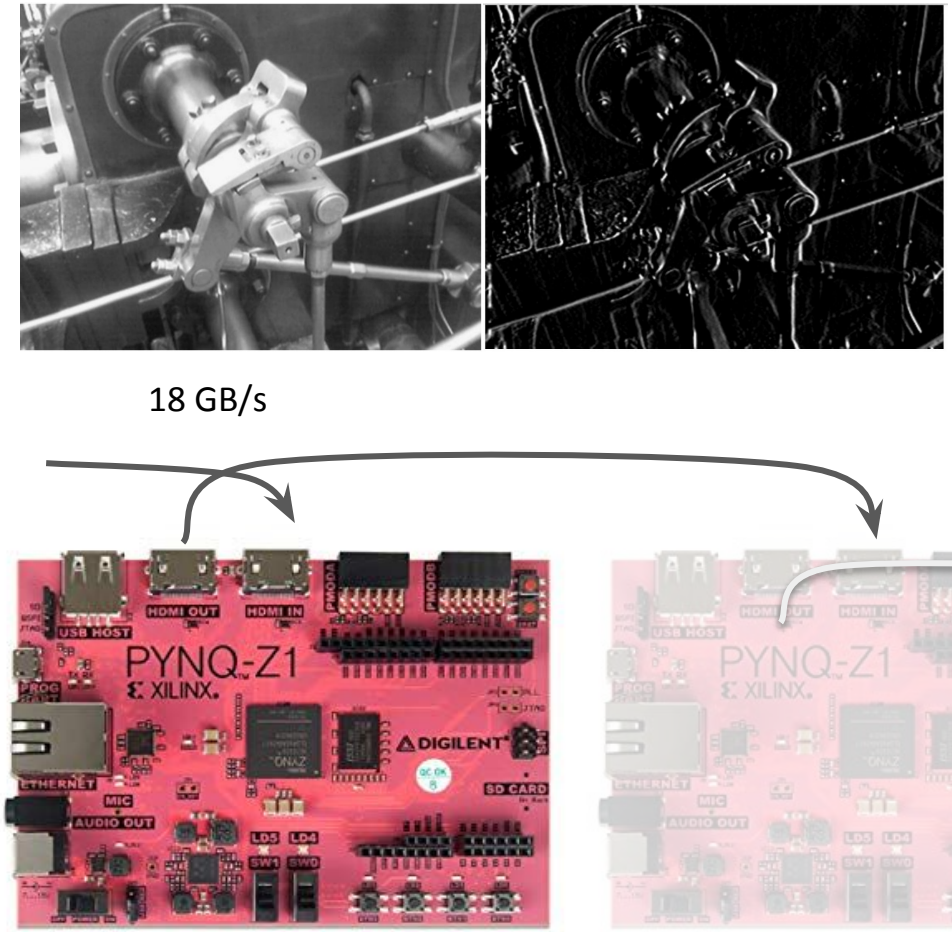
Automatic conversion of high-level ML-specific code into Xilinx-specific code.

OpenCL APIs for (some) Altera FPGAs to serve as alternative to GPU as hardware accelerators

FPGAs for fixed latency, scalable inference

Several layers of a large and deep neural network can be splitted across different FPGAs, connected with high throughput cables.

This allows to obtain fixed-latency evaluation of extremely complex operations possibly emulating the result of fits or other iterative operations that are “emulated” by the neural network.



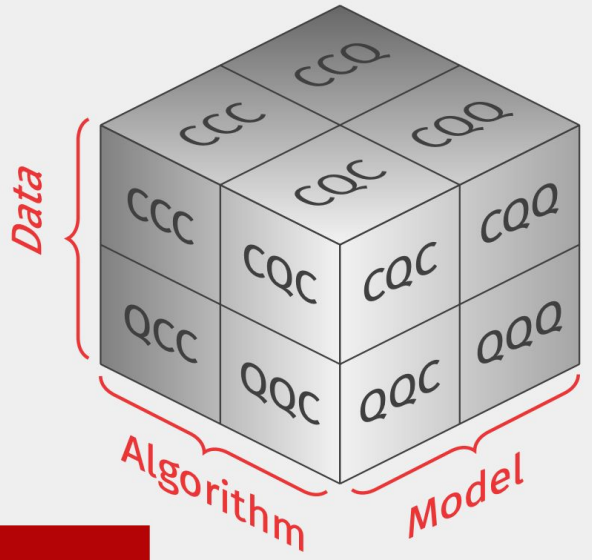
Dedicated hardware available for further studies in Florence (thanks LHCb)

Machine Learning and Quantum Computers

Quantum Computing can provide helpful insights in Machine Learning

AND

Machine Learning can be used to model Quantum Computation.



Problem in ML

effectively model a complex mapping between huge dimensional spaces using few trainable parameters

Analogy: variational methods for quantum many-body systems

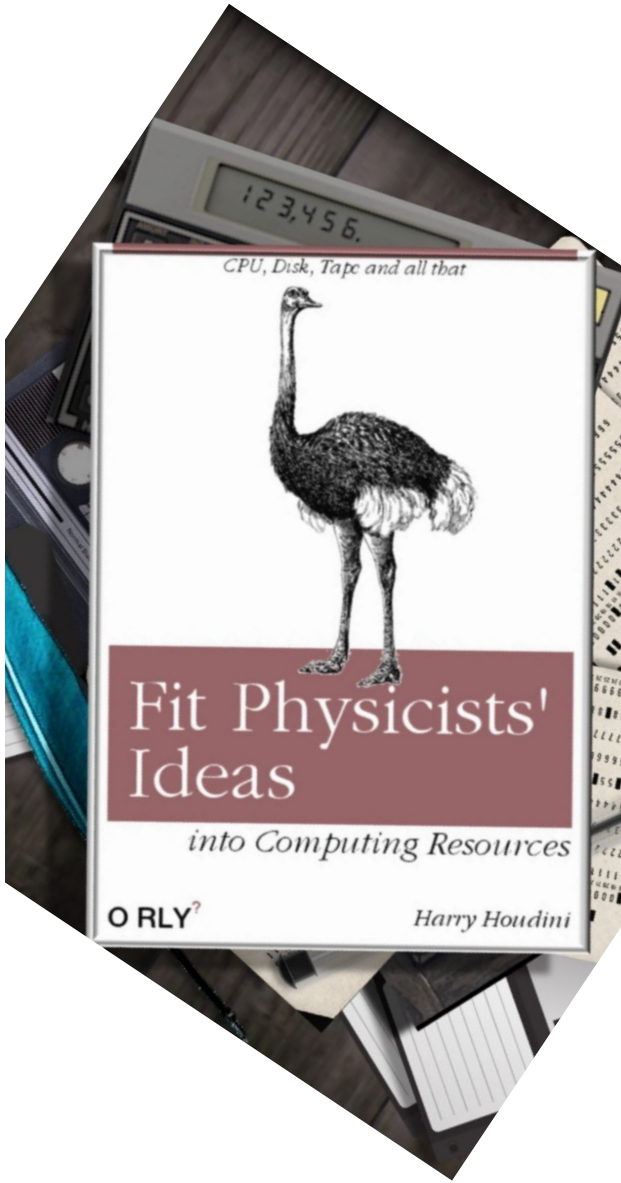
$$E_{\rm GS} \lesssim \min_{\theta} \langle \psi(\theta)| H |\psi(\theta)\rangle$$

IBM rigetti IONQ XANADU Google AI Quantum

Early quantum computers are available to solve machine learning problems through quantum algorithms or quantum annealing.
Though it is not clear it is convenient wrt. CPUs and GPUs. Probably yes.

Conclusion

- Machine Learning ***is*** fitting
- Historically, nuclear and subnuclear particles played a role in the development of **statistical methods for the interpretation of large datasets.**
- Today we welcome a **renewed interest** from other scientific communities and **private companies** driven by the ability of “fitting” images (photographies, videos).
- Handling larger amounts of data is a recurrent requests in experimental physics, we should just not miss the opportunity.



Groups within **INFN Firenze** are active on several very interesting developments and backed by **national initiatives**. An important thrust from sharing code and data, made possible by the ***Servizio Calcolo e Reti*** (thanks!)

Training, Scientific and Outreach Events (random order)

- Tesi di Laurea in Ingegneria Informatica su sviluppo di trigger per HERD (Nicola Mori)
- Corso per Dottorato in Fisica e Astronomia "Tecnologie Computazionali Moderne per la Fisica delle Alte Energie"
- Workshop "Artificial Intelligence in Medical Physics" (2019-09-11, Cinzia Talamonti)
- Caffé Scienza sul tema dell'Intelligenza (Osteria del CUS, 2019-05-07)
- Seminari per Art&Science across Italy (Reti Neurali Creative)

Art & Science ACROSS ITALY

Signed an agreement with the Tuscany Ph.D. Program on ***Smart Computing*** *https://smartcomputing.unifi.it*

Planned a *First International Workshop on Machine Learning in Physics for October, 2020*

More news soon

UNIVERSITÀ DEGLI STUDI FIRENZE | Dipartimento di Fisica e Astronomia

UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO Dipartimento di Ingegneria dell'Informazione