6th International Workshop
on Deep Learning in Computational Physics
MLIT, JINR, Dubna, July 6-8, 2022

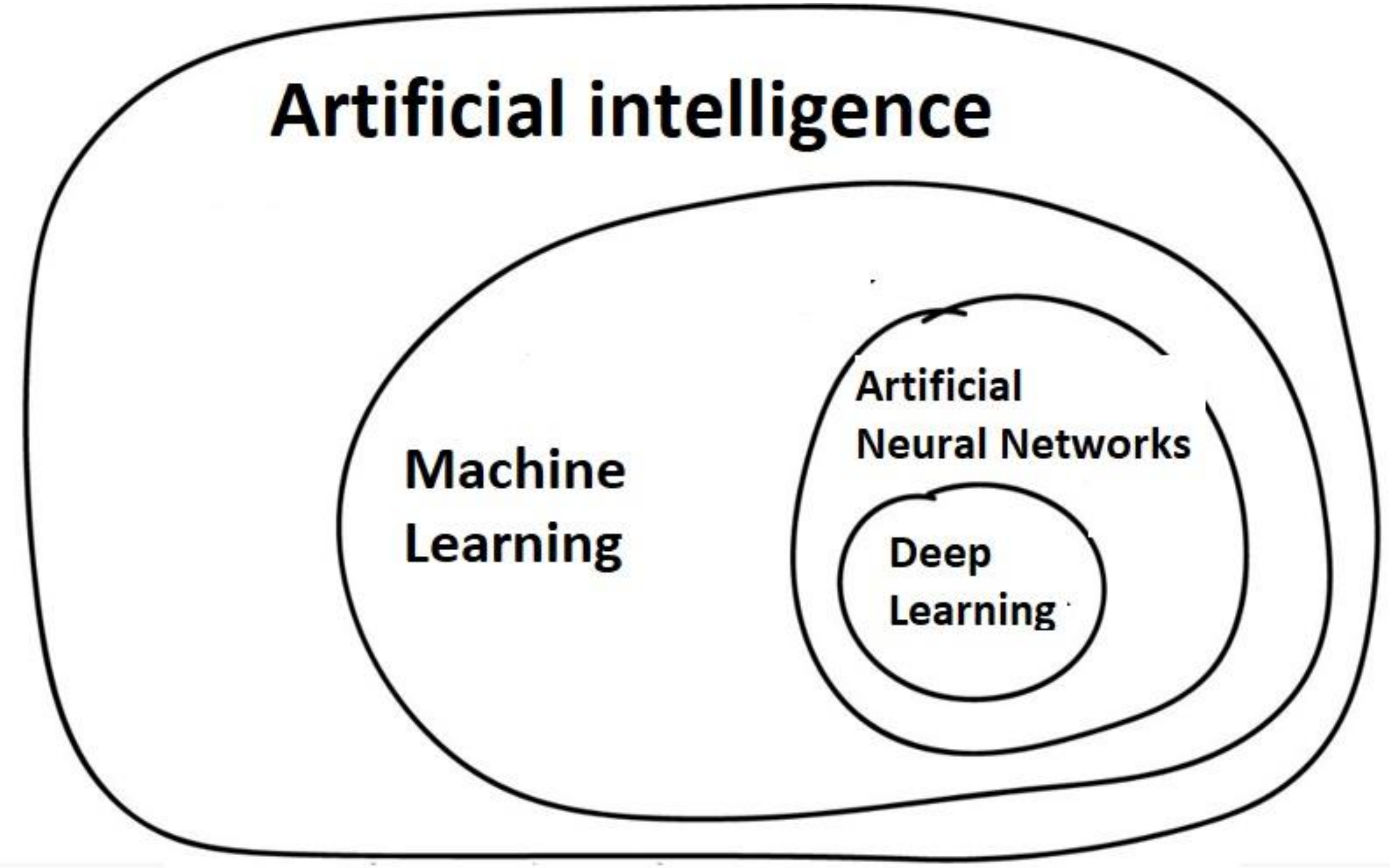
Machine Learning in MLIT.

History, Challenges, and Prospects

Gennady Ososkov

Laboratory of Information Technologies,
Joint Institute for Nuclear Research
141980 Dubna, Russia
email: ososkov@jinr.ru
http://gososkov.ru

Artificial intelligence -
- Machine Learning -
- Deep Learning



Machine Learning

Machine learning (in a sense a computer learning) is a type of artificial intelligence that provides computers with the ability not just use a pre-written algorithm, <u>but to teach themselves how to solve a problem using a large data sample in order to recognize and adapt when exposed to new data.</u>

Machine Learning (ML) Methods can be divided into classical and modern ones

<u>Classical ML methods:</u>

- **regression (linear, nonlinear and <u>robust</u> fitting of univariate and multivariate sets) ;**
- **classification as a variant of pattern recognition (decision trees, support vector machines, Genetic and swarm algorithms, Hough transform);**
- **dimensionality reduction (PCA);**
- **clustering (K-means).**

Modern ML methods due to recent appearance of neural networks with deep learning: multilayer perceptrons, autoencoders, convolutional and recurrent neural networks.

In the following narrative of the history and current status of physical experiment data processing, I will emphasize how machine learning techniques have been used to create processing algorithms, evolving along with technological advances

History of ML in MLIT 1. Very beginning

M.G.Meshcheryakov

Machine learning (ML) methods began to be used in the MLIT laboratory from the very beginning of its organization in **1966**, when one of the main tasks of the LVTA was the **automation of film data processing** used at that time in **physics experiments with bubble chambers with manual measurements**

Tracking was done manually during measuring points on different projections. Then the matching procedure used to obtain space points for every track. To these points a helix corrected for nonhomogeneous magnetic field was fitted to get the particle parameters. The fitting could already be done with a program Fumili for minimizing nonlinear functions, written by Igor Silin, first in M-20 machine codes, and then rewritten by him in Fortran.

The vertex of the event was sought by fitting as the point closest to all the tracks.

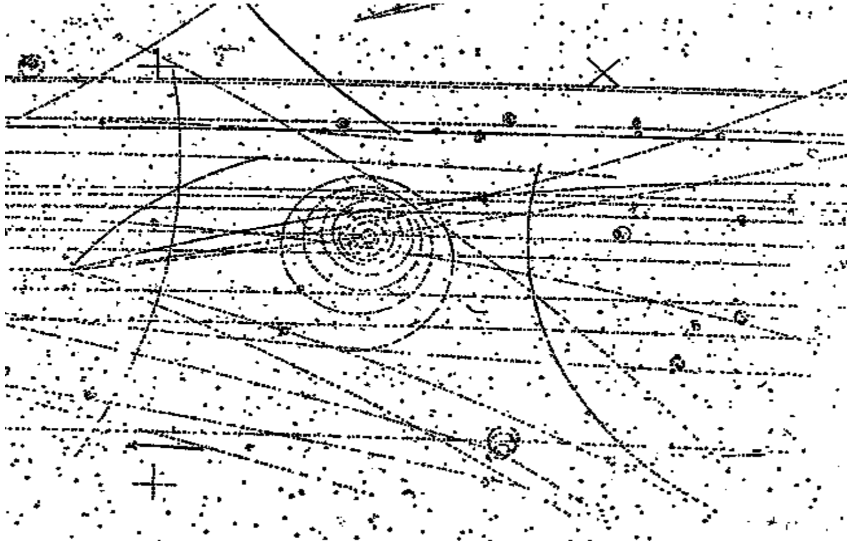
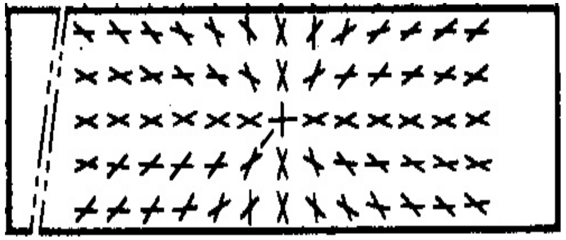


Photo of an event projection of the CERN 2m hydrogen bubble chamber.
P-P interaction with few tracks,
but many beam tracks and noise points

History of ML in MLIT 1. Era of automatic scanners

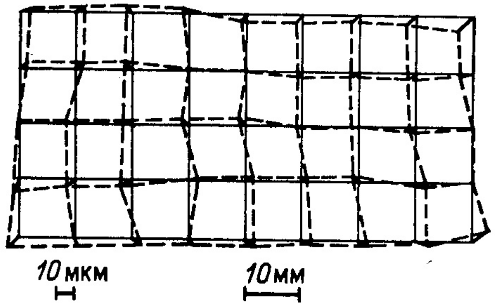
"Spiral Reader" scanner, in which the operator put a point at the vertex of the event, from which the image was scanned along a spiral **by a light spot in the form of a thin slit.** All these scanners use light spot for scanning with inevitable optical distortions. Therefore a **calibration procedure was needed**

Calibration

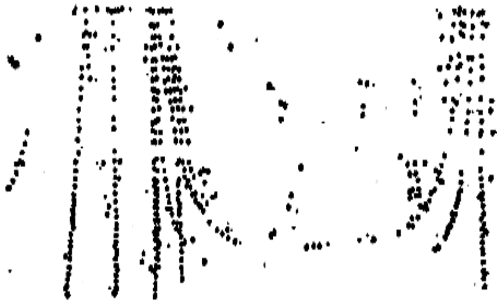


reference plate

Calibration measurements of reference plate with specially placed crosses was scanned. The pattern recognition method was used to recognize crosses. Their centers were calculated and compared to their ideal positions to develop a map of residuals, with the help of which a calibration transformation was constructed that should eliminate optical distortions in measurements.



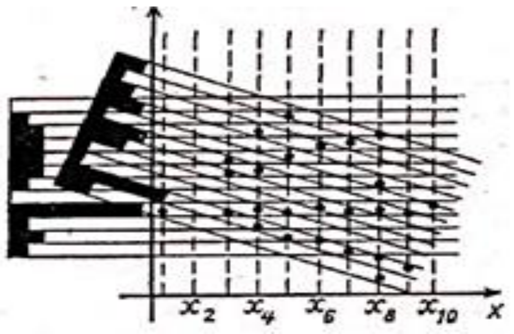
map of residuals



Scanned event projection in polar coordinates

Tracking for different projections was a more complicated pattern recognition problem required a prior conversion scanned data to polar coordinate system in which the tracks were "straightened" so that they could be found by the Hough transform.

The vertex of the event was also sought by fitting as the point closest to all the tracks.

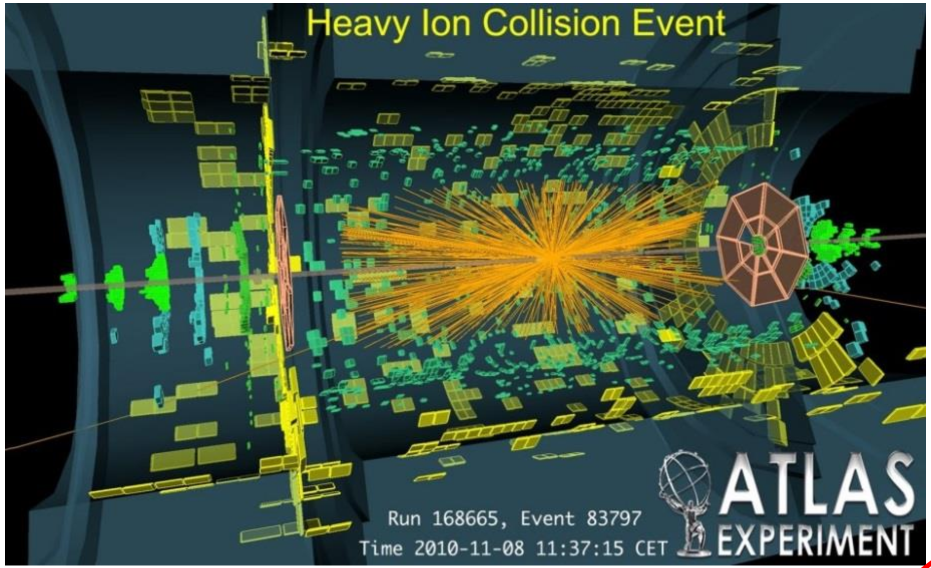


Hough transform variant as rotating histograms

The experimental data processing tasks for these experiments, which have already passed into history, were solved by classical methods of machine learning using the then available algorithms of pattern recognition, curve and surface fitting, using Monte Carlo models to compare the experimental results with physical theory. Many of the algorithms invented then proved useful in later years as well

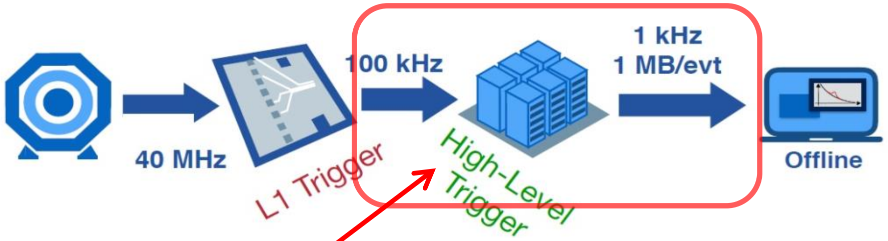
ML in MLIT. Entering era of Electronic Experiments

Large Hadron Collider – LHC at CERN



ATLAS – is one of four LHC experiments

An example of one HEP experiment



Scheme of the experimental data processing

rate now about 1 petabyte/sec
exabytes of data per year

1 petabyte=10^{15} bytes
1 exabyte=10^{18} bytes

RUN 1 2012 Higgs boson obtained!

RUN 2 2015-2017 - already <u>exabytes</u> of data per year!
All four LHC experiment detectors provide data at a rate of about 1 petabyte / sec.

A system of intelligent triggers and filters compresses this data millions of times, leaving only useful information for long-term storage. As a result, the <u>LHC in RUN2 generated 50 terabytes per second</u> for storage - as much data in 4 hours as the <u>entire Facebook network collects in a day</u>.

The growing role of <u>Monte Carlo simulations of the events under study</u>, taking into account all the details of the experimental setup

It is impossible to process such a volume of data at CERN, therefore

1. Created the Worldwide LHC Computing Grid (WLCG).
2. Numerous software packages have been developed for data modeling and analysis using machine learning methods

Main stages of data analysis in HEP experiments

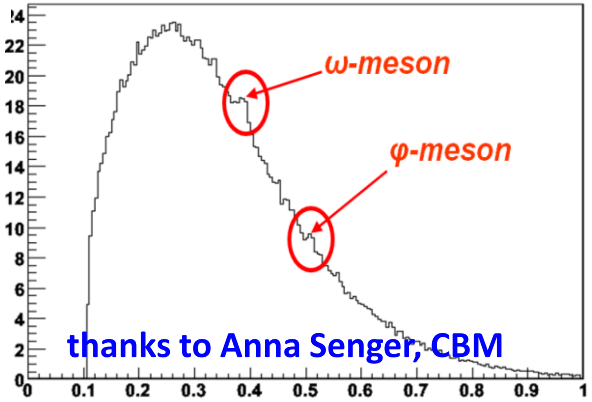
- Collect data from many channels on many sub-detectors (terabyts/sec)
- Decide to read out everything or throw event away (multilevel trigger system, some of them used the ML technique to prevent loss of useful events)
- Build the event (put information together)
- Store the data
- Analyze them
  - correct data for detector effects: calibration, alignment
  - hit detection, tracking, vertex finding, revealing Cherenkov rings ,
  - fake objects removing
  - user analysis algorithms
  - data volume reduction
- do the same with a simulation

  correct data for detector effects etc
- Compare data and theory

  Invariant mass spectroscopy for short-lived particles and resonances

<u>Applied ML methods</u>

- Hough transforms,
- cellular automata,
- Kalman filter,
- robust estimation,
- wavelet analysis, etc.
- artificial neural networks,



New inevitable issues of the WLCG and distributed computing era

- Parallel programming of optimized algorithms
- Grid-cloud technologies which changed considerably HEP data processing concept

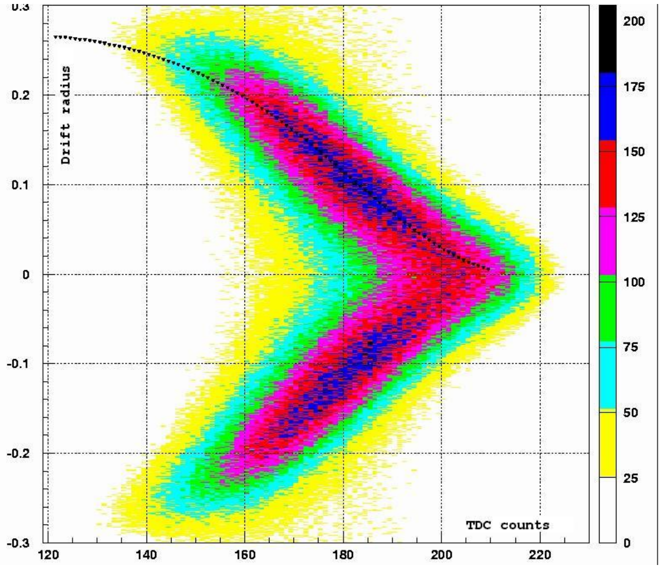
ML in MLIT examples

1. Calibration of drift chambers

An example for HERA-B OTR drift chamber calibration:

Drift time measured in TDC (Time-Digital Converter) counts is to be transferred into **drift radii** to obtain the calibration function ***r(t)***

r(t) **was obtained by robust fitting cubic splines to directly 2D histogram with many thousand bins**



drift radii versus TDC counts

2. Alignment problems.

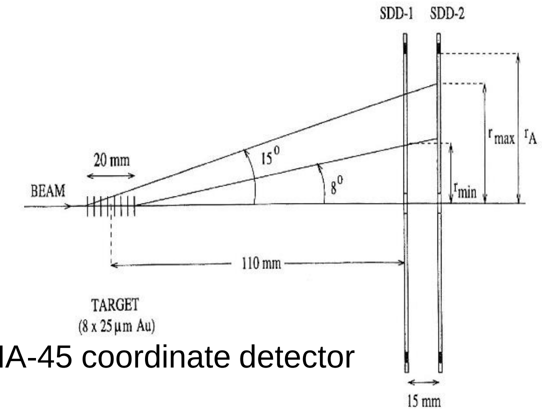
Electronic detectors consist of many thousands of modules, which, when even carefully assembled into an experimental setup, are inevitably subjected to various minor disruptions, such shifts and inclinations, distorting measurement results

Therefore, an alignment procedure is needed to detect and remove possible distortions. Misalignment of detector modules can be found by analyzing residuals between the measured values and fitted track coordinate. These residuals are functionally dependent on the two types of parameters: track model parameters and alignment parameters. So the alignment problem can be formulated as a mathematical problem of minimizing a functional summing the squares of all residuals by both types of parameters. The intrigues is that the number of those parameters reaches millions. The solution has been proposed to use singular value decomposition technique to diagnose these unconstrained degrees of freedom.

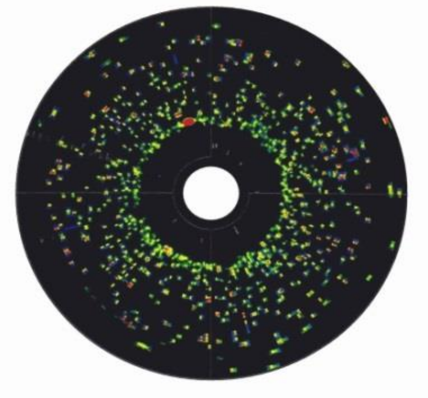
As the result we have Data-driven alignment!

3. Robust fitting application examples

1. NA-45. Determination of the interaction vertex for only two coordinate planes



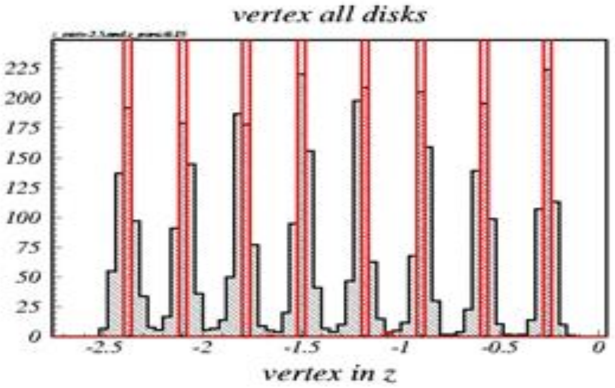
NA-45 coordinate detector



One of 2 silicon disks

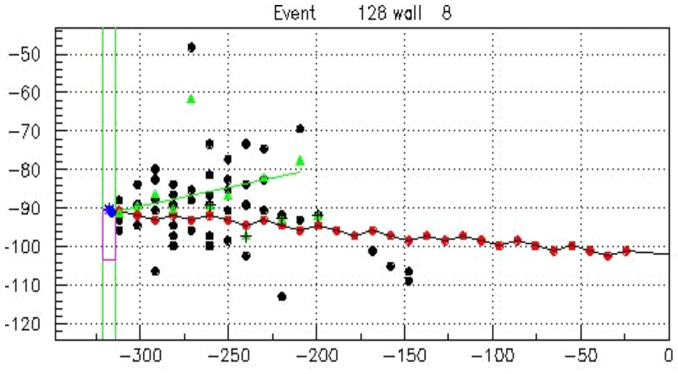
The target consists of eight 25-***μ*** gold discs. **700 track** events **in narrow angular acceptance** and large number of noise counts did not allow to recognize individual tracks.

Robust fitting iterations converged in **five iterations**, although initial approximation was roughly taken as the middle of the target region.

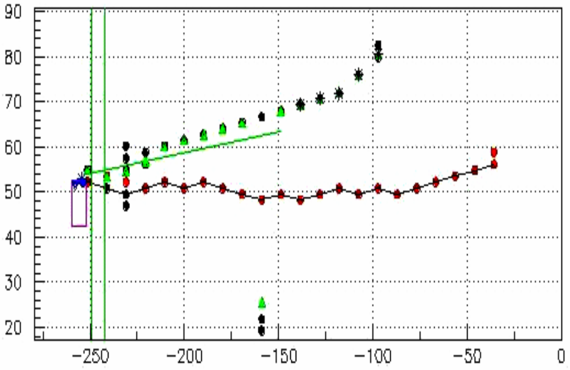


2. Opera. 2D weights for hadron showers and muon tracks.

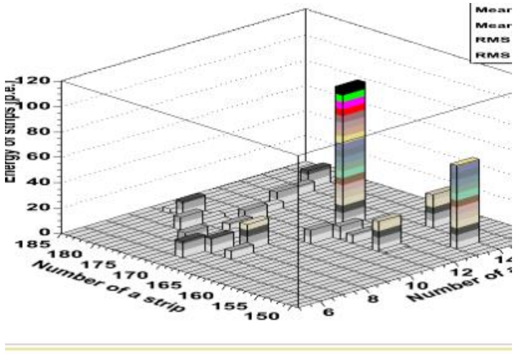
Fitting with 2D robust weights, which depend not only on distance of a point to the fitted track, but also on amplitudes of track hits



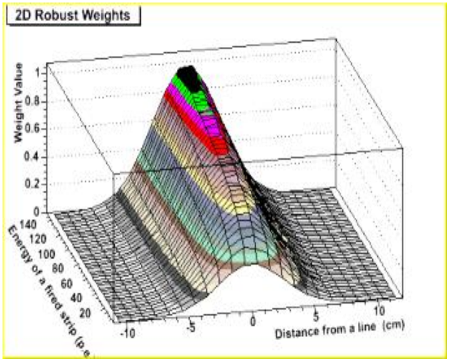
hadron shower



1D muon tracks



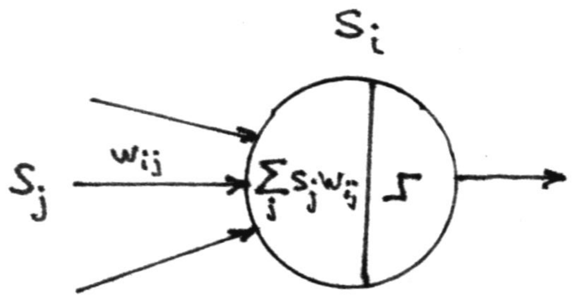
2D muon tracks



2D weight

Artificial Neural Networks. Formalism and types

Artificial neuron

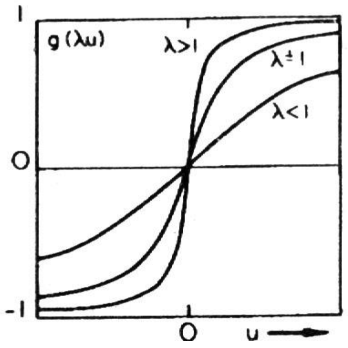


Connection between i^{th} and j^{th} neurons is characterized by synaptic weight w_{ij}

the *i*-th neuron output signal

$$h_i = g(\Sigma_j w_{ij}\, s_j)$$

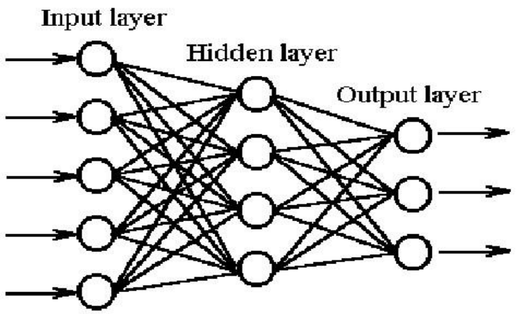
Activation function g(x). As usual, it is sigmoid $g(x)=1/(1+exp(-\lambda x))$, but not only



There are many ways to combine artificial neurons into a neural network.

Main types of neural nets applied in HENP in the recent past

1. Feed-forward ANN. If there is one or more hidden layers it names as Multilayer Perceptron (MLP)

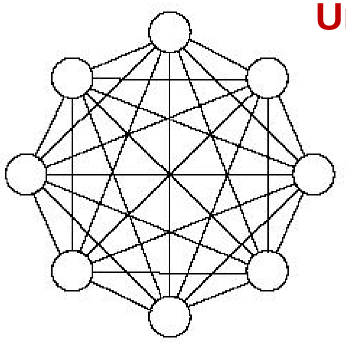


Supervised learning

The purpose of training is to determine weights so that the trained network solves the problem of recognition or classification

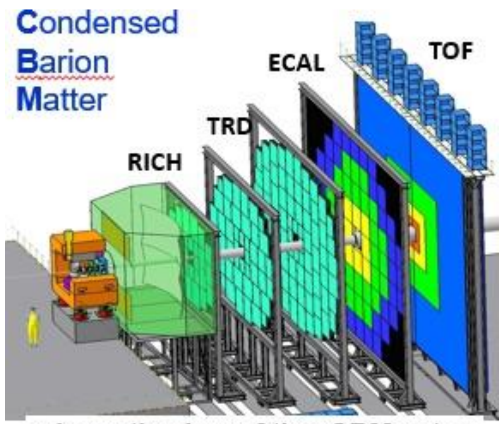
2. Fully-connected recurrent ANN (Hopfield net).

Unsupervised self-learning



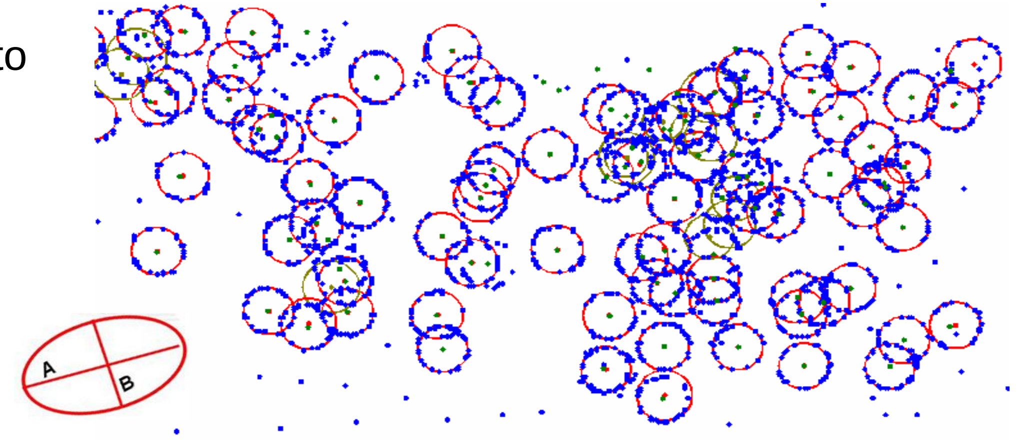
Further, you will learn about new types of neural nets that are becoming more increasingly applicable in HENP

4. MLP application example: CBM RICH detector

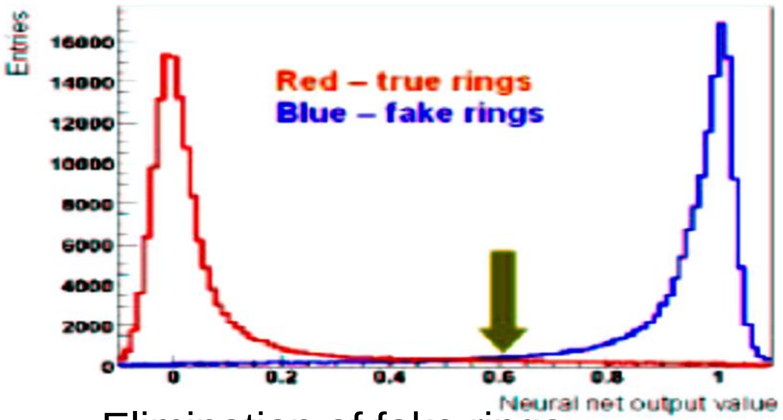


schematic view of the CBM setup

RICH produces many Cherenkov radiation rings to be recognized with evaluating their parameters despite of their overlapping, noise and optical shape distortions.
In order to **distinguish between good and fake rings** and to **identify electron rings** the study has been made to select the **most informative ring features**.

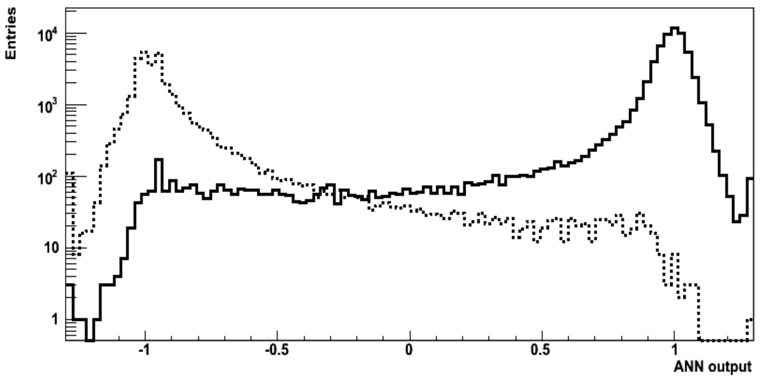


10 of them have been chosen to be input to ANNs, such as 1.number of points in the found ring, its distance to the nearest track, χ^2 of ellipse fitting, both ellipse half-axes (A and B) etc.



Elimination of fake rings

Two samples with 3000 e (+1)and 3000 π (-1) have been simulated to train both ANN. Electron recognition efficiency was fixed on 90%
Probabilities of the 1-st kind error 0.018 and the 2-d kind errors 0.0004 correspondingly were obtained

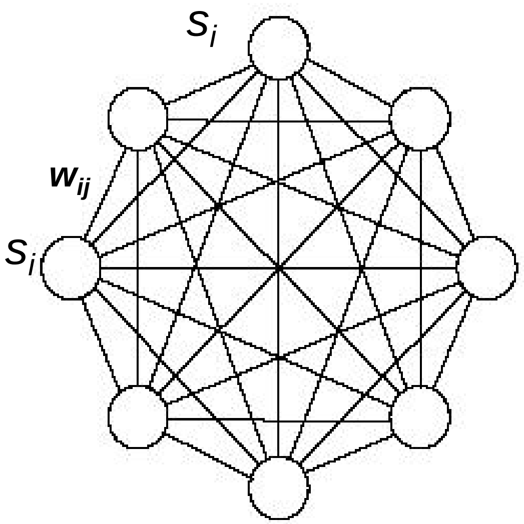


Identification of electron and pion rings

Cherenkov ring recognition is the part of the more general event reconstruction problem as for CBM, as for other experiments

Fully connected recurrent neural networks

The recurrent fully connected NN considered as a dynamic system of binary neurons. $s_i = \begin{cases} 1, active \\ 0\ , non\ active \end{cases}$



All them are connected together with weights w_{ij}.

Hopfield's theorem: the energy function

$$E(s) = -\tfrac{1}{2} \Sigma_{ij}\, w_{ij}\, s_i\, s_j$$

of a recurrent NN with the symmetrical weight matrix

$w_{ij} = w_{ji}$, $w_{ii} = 0$ has local minima corresponding to NN stability points.

However the usual way of $E(s)$ minimizing by updating the equation system, which defines the ANN dynamics: $s_i = \frac{1}{2}\left(1 + \mathrm{sign}\left(-\frac{\partial E}{\partial s_i}\right)\right)$ would bring us to one of many local minima.

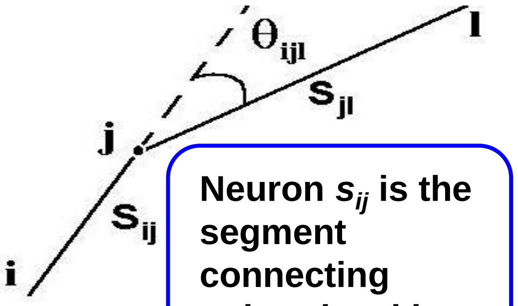
Since our goal is to find the global minimum of ***E*** we have to apply the **mean-field-theory (MFT).** According to it, **all neurons are thermalized** by inventing temperature T, as λ=1/T and s_i are substituted by their thermal averages $v_i = < s_i >_T$, which are continuous in the interval [0,1]. Then ANN MFT dynamics is determined by the updating equation $v_i = 1/2(1+\tanh(-\partial E/\partial v_i\ 1/T) = 1/2(1+\tanh(H_i /T)$,

where $H_i = <\Sigma_j w_{ij}\, s_j >_T$ – is the local mean field of a neuron. Values of v_i are now defined **the activity level of i^{th} neuron**. Neurons with $v_i > v_{min}$ determine the most essential ANN-connections

5. Hopfield NN (HNN) applications in HENP

Track recognition by Denby- Peterson (1988) segment model

The idea: support adjacent segments with small angles between them. It is done by the special energy function:



Neuron s_{ij} is the segment connecting points *i* and j

$E = E_{cost} + E_{constraint}$, where

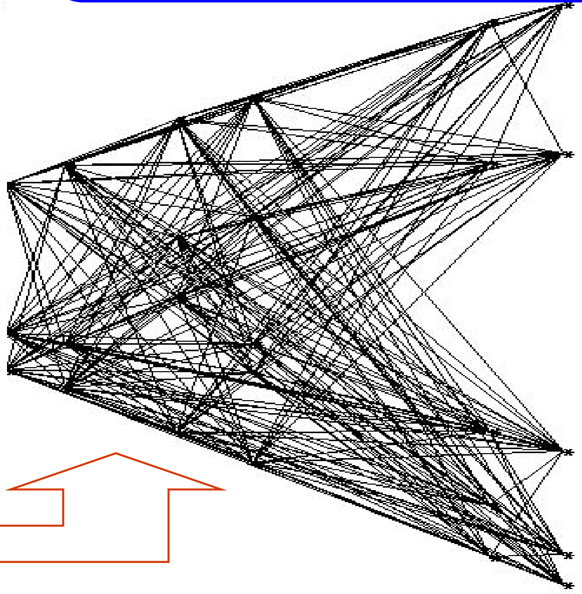
$$E_{cost} = -\frac{1}{2}\sum_{ijkl} \underbrace{\delta_{jk}\frac{\cos^m \theta_{ijl}}{r_{ij}r_{jl}}}_{w_{ijkl}} v_{ij}v_{kl},$$

An easy example of the EXCHARM experiment with 6 multiwire proportional chambers registering a lot of hits from passing particle tracks and noise

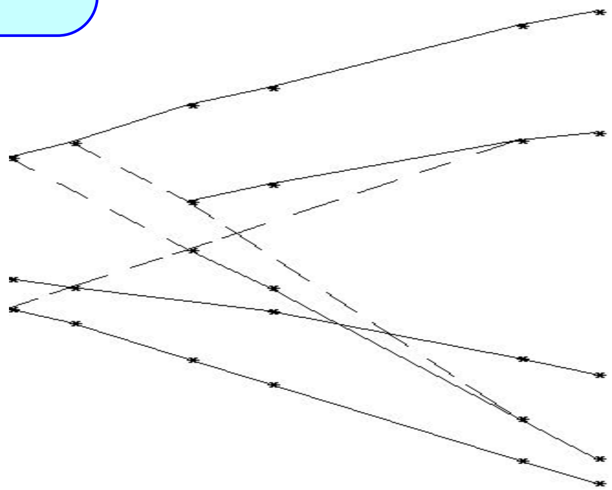
$E_{constraint}$ punishes segment bifurcations and balances between the number of active neurons and the number of experimental points.

Note: **adding even a single noise point would generate**

~80 extra hampering neurons



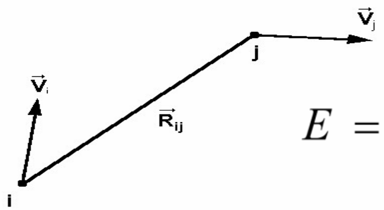
Zero iteration: 244 neurons.



After 30 iteration:
26 neurons with $v_{ij} > 0.5$

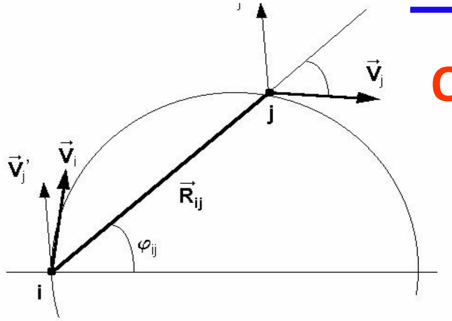
A more advanced type of HNN, called "Elastic NN" allows you to get rid of dependence on noise and, most importantly, to combine both event reconstruction stages: recognition and fitting tracks in the entire event or each track individually

Track recognition by rotor models of Hopfield networks



$$E = -\frac{1}{2}\sum_{ij}\frac{1}{|r_{ij}|^m} v_i v_j - \frac{1}{2}\alpha\sum_{ij}\frac{1}{|r_{ij}|^m}(v_i r_{ij})^2$$

The energy function: the first term forces neighbouring rotors to be close to each other. The second term is in charge of the same between rotors and track-segments.



Our innovations to rotor's approach

Thanks to V.Kisel

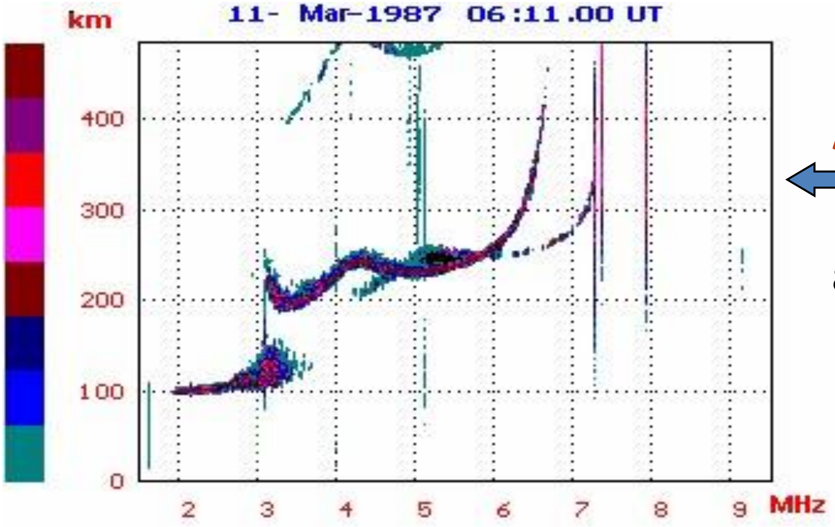
$$v'_j = \begin{pmatrix} \cos 2\phi_{ij} & \sin 2\phi_{ij} \\ \sin 2\phi_{ij} & -\cos 2\phi_{ij} \end{pmatrix} v_j = W_{ij} v_j$$

Therefore we obtain a simple energy function without any constrains $E = -\frac{1}{2}\sum_{ij} v_i \cdot v'_j$

ARES experiment demanded extra efforts:

- **prefiltering by cellular automaton;**
- **local Hough algorithm for initial rotor set up;**
- **special robust multipliers for synaptic weights.**

Results: recognition efficiency - 98%



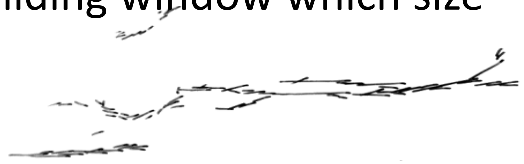
Analysis of ionograms.

Data from the vertical sounding of the ionosphere are measurements of frequency, amplitude, and arrival time of a signal reflected from an ionospheric layer.

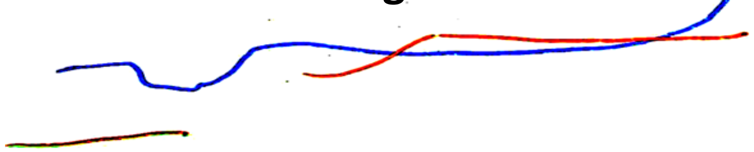
Stages of ionogram processing: 1. Cellular automaton removed vertical noise and filled in missed hits

2. Rotors were initiliazed by angular histogramming in a sliding window which size was varied depending on the track curvature.

It decreased the neural net evolution up to 3-5 iterations



3. Recognition result

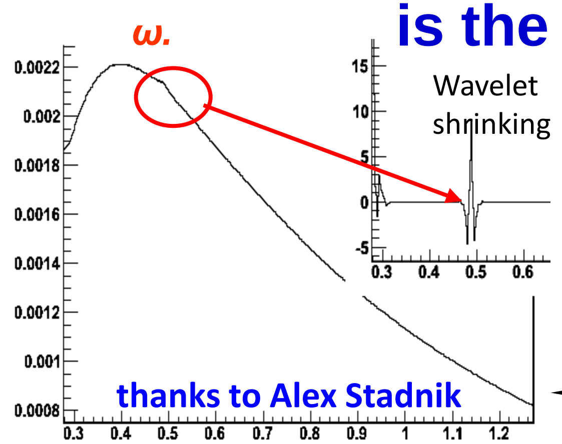


Up to now the corresponding program is in use in the Irkutsk Institute of the terrestrial magnetism, Russia and in the Lowell University, USA

Wavelet transform

is the way to go to domain, where ML problems could be solved much easier

we need them for handling invariant mass spectra when **S/B ratio is << 1**



1. Discrete wavelets

for resonance indicating even in presence of massive background

Continues wavelet G_2 transforms a gaussian $g(x;A,x_o,\sigma)$ into wavelet of the same

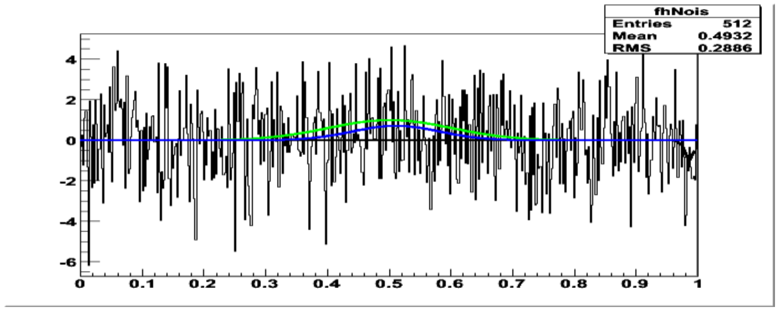
$$W_{G_2}(a,b)g = \frac{Aa^{5/2}\sigma}{(a^2+\sigma^2)^{3/2}} G_2\left(\frac{b-x_0}{\sqrt{a^2+b^2}}\right)$$

order, but with parameters of that gaussian:

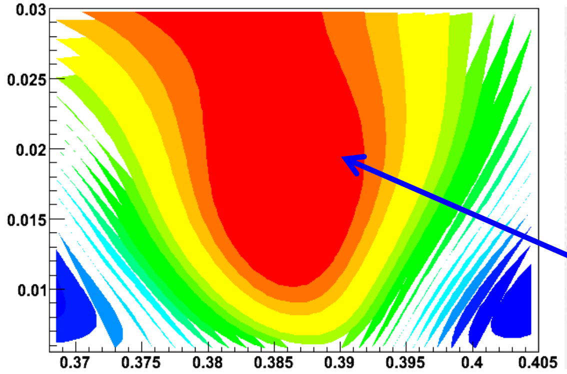
It is true for any order n and leads to the idea of looking for the peak parameters directly in G_2 domain <u>without its inverting</u>

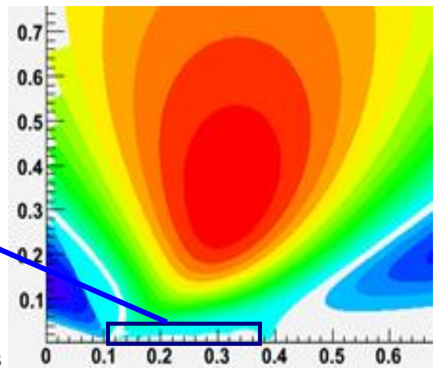
Continues wavelets are non-orthogonal , thus they cannot be inverted , therefore nobody uses them

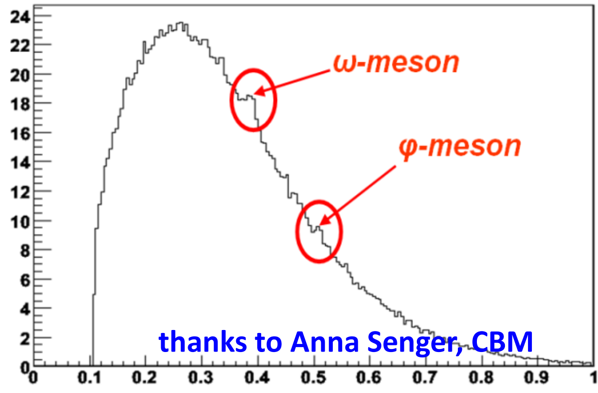
2. Smoothing after background subtraction without losing any essential information



3. evaluating peak parameters from invariant mass spectrum





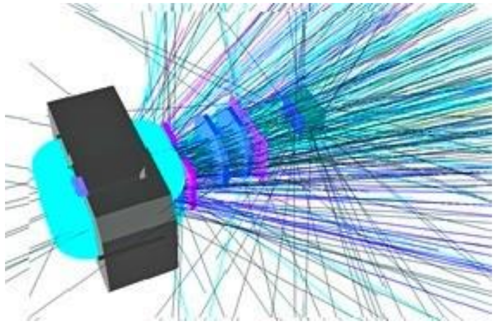


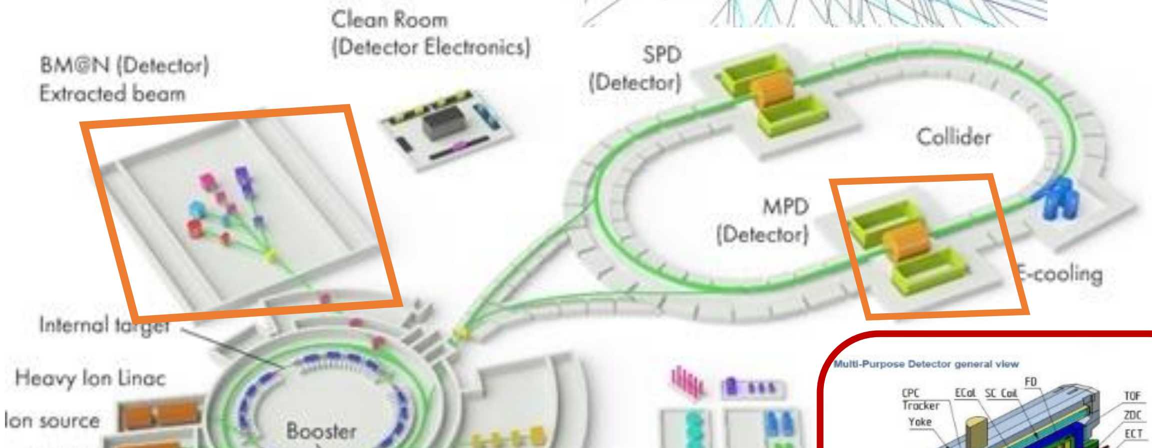
Challenges of the modern time and arising problems

1. The main source of data comes from experiments in High Energy Physics (HEP)

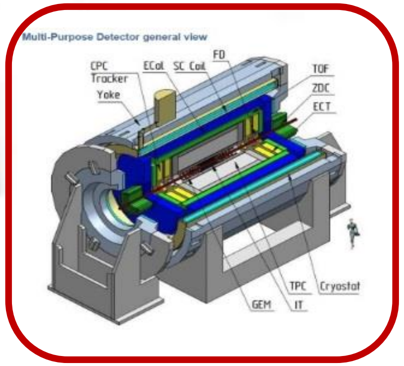
NICA Megaproject at JINR

Experiment BM@N
Strip GEM detector inside the magnet.
Simulated Au+Au event is shown





Scheme of the NICA complex with experiments BM@N, MPD, SPD

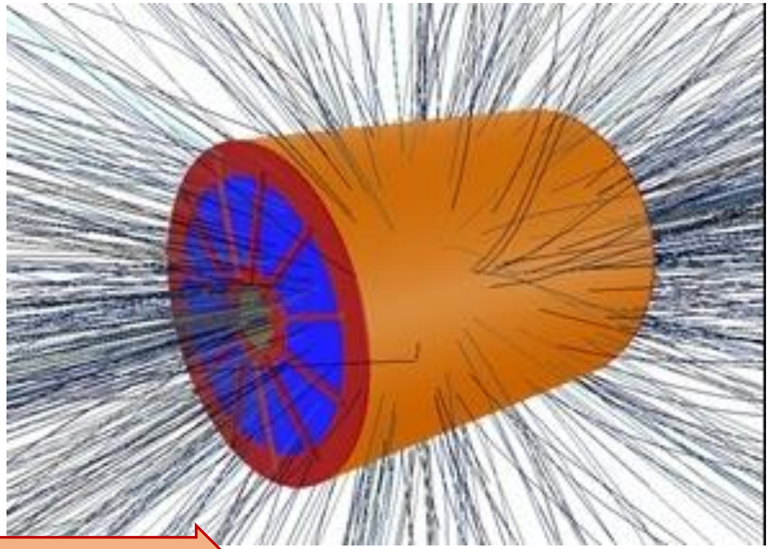


There are two types of HEP experiments - with fixed target (BM@N) and collider ones (MPD)

NICA performance forecast

Data transfer rate: 4.7 GB / s 19 billion events per year. After processing and analysis it gives 8.4 PB per year for storage

TPC track detector inside MPD magnet.
Simulated event from the interaction of gold ions, generating thousands of tracks is shown

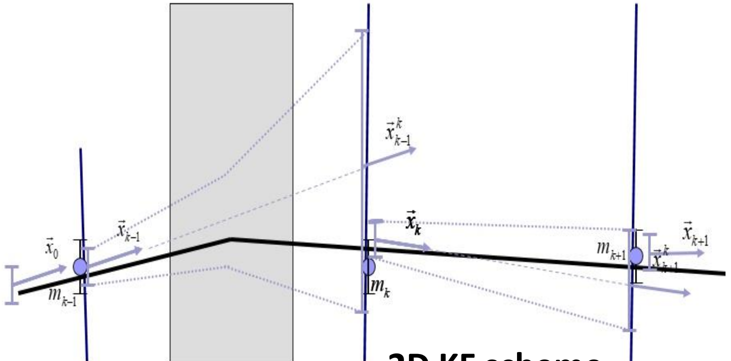


Challenge of Deep Learning approach in Neural Networks, neccessity to use parallelism and virtuality.

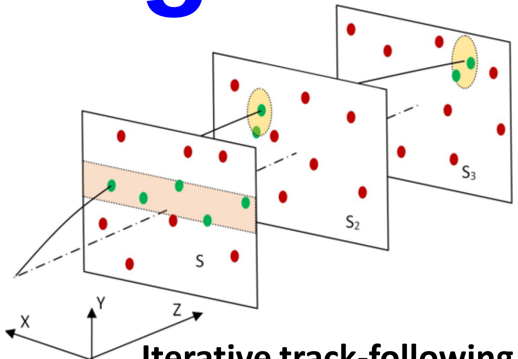
From Kalman filter to deep tracking

The Kalman filter (KF) is an efficient recursive filter that estimates the state of a linear dynamical system using a series of imprecise measurements.

The state vector $\vec{x} = (x, y, t_x, t_y, q/p)^T$ is iteratively evaluated to predict the position of the track on the next coordinate plane taking into account changes in the covariance matrix and error corridors.



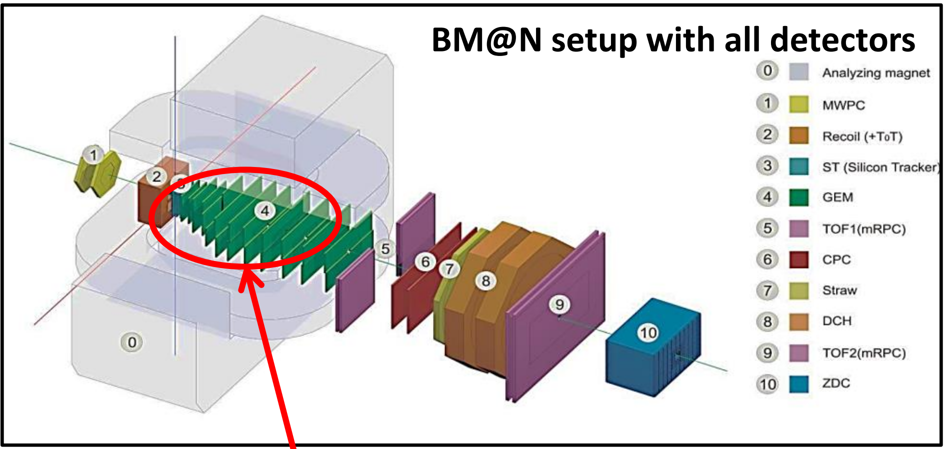
2D KF scheme

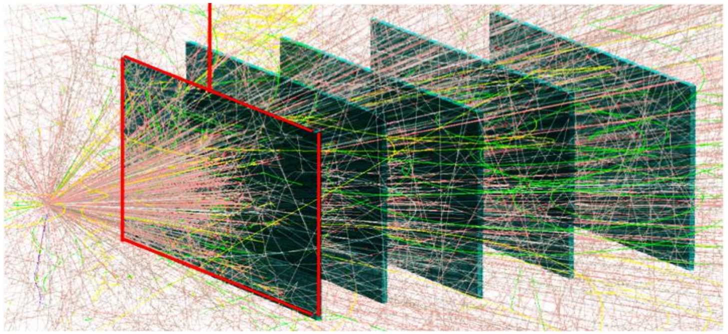


Iterative track-following

- KF allows one to take into account the inhomogeneity of the magnetic field, multiple scattering and energy losses when a particle passes through the detector medium. KF extrapolates the initial state of the track (as a rule, the first three points) to a small area on the next coordinate surface. **Therefore tracking algorithms based on KF have been used traditionally with great success in HEP experiments for years.**
- However, the **initialization procedure** needed to start Kalman Filtering requires a tremendous search of hits aimed to obtain so-called “seeds”, i.e. initial approximations of track parameters of charged particles.
- Besides these techniques are **inherently sequential and scale poorly** with the **expected increases in detector occupancy to the exabyte level** in new conditions as for modern and planned experiments as HL-LHC or NICA.
- **Thus, a new approach to the reconstruction of physical events observed in experimental detectors is inevitably required. It should be implemented by new, an order of magnitude faster, parallel and highly scalable algorithms.**
- **Deep learning algorithms bring a lot of potential to this problem due to their capability to model complex non-linear data dependencies, to learn effective representations of high-dimensional data through training, and to parallelize on high-throughput architectures such as GPUs.**

Baryonic Matter at Nuclotron (BM@N)

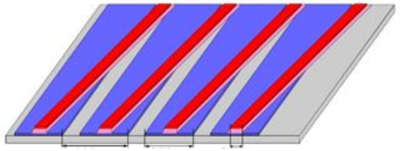




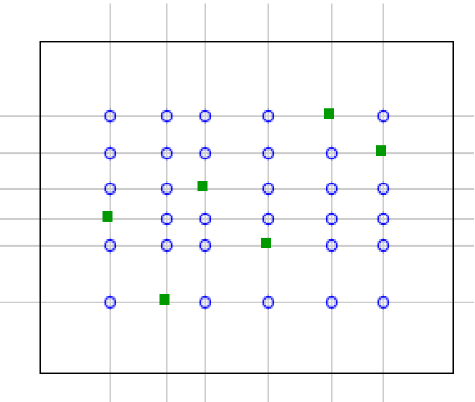
Visualization of simulated Au+Au event

Our problem is to reconstruct tracks registered by the GEM vertex detector with 6 GEM-stations (RUN 6, spring 2017) inside the magnet.

Problems of microstrip gaseous chambers



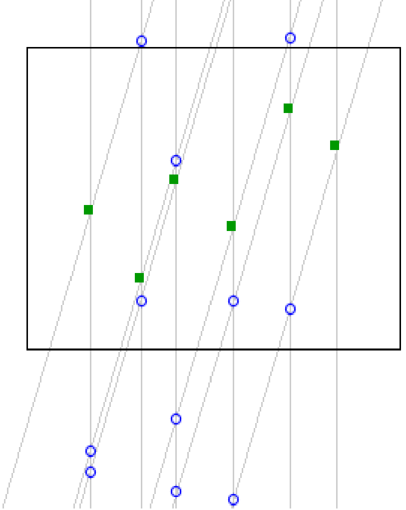
The main shortcoming is the appearance of fake hits caused by extra spurious strip crossings. For n real hits one gains n^2- n fakes

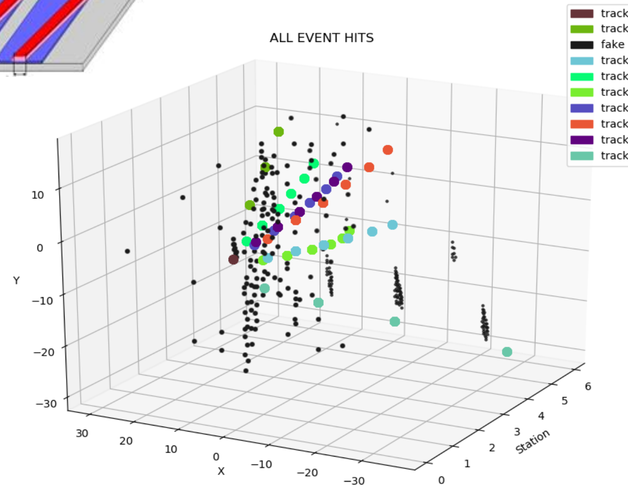


- Real hit (electron avalanche center)

- Spurious crossing

One of ways to decrease the fake number is to rotate strips of one layer on a **small angle** (5-15 degrees) in respect to another layer





3D image of the C+C event with 9 tracks. Black dots are fakes

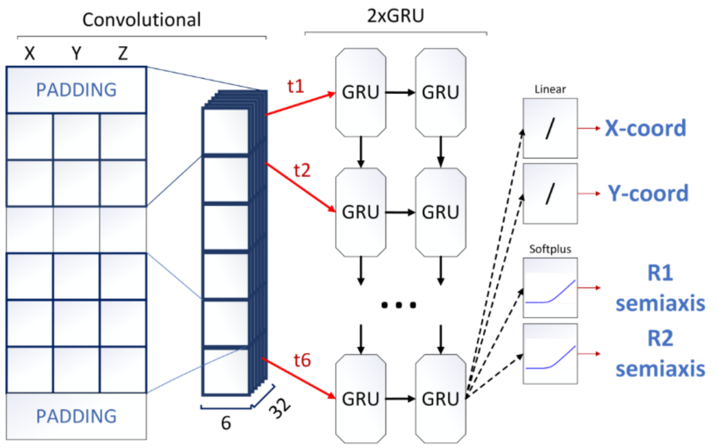
Although small angle between layers removes a lot of fakes, pretty much of them are still left

Local and global tracking approaches

There are two approaches to implement any tracking procedure:

1. **Local tracking, when tracks are reconstructed one after the other**, as in KF algorithm.
   Drawbacks,- slow, no way to see the dependence between individual tracks or groups of tracks and such phenomenon as secondary vertices, the need to realize the next stage for looking the even vertex.
2. **Global tracking**, in which the recognition of tracks among noises is performed **immediately across the entire picture of the event**

1. Local tracking for the GEM detector of the BM@N experiment is especially difficult because of the giant number of fake hit presence, which makes it extremely difficult to search for those hits at subsequent detector stations that are a continuation of the track being processed.

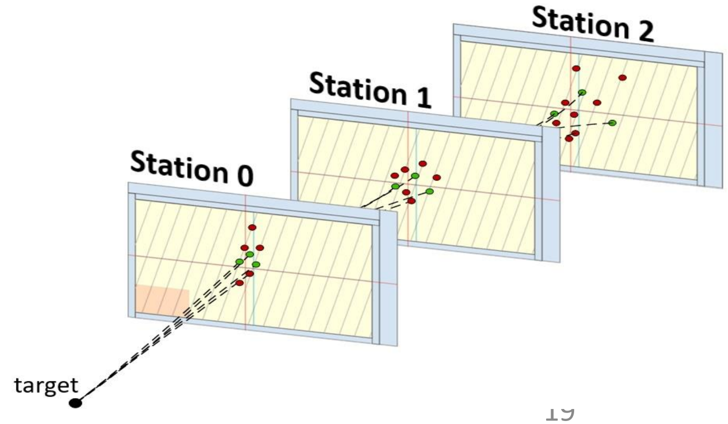


Scheme of the recurrent TrackNETv2 neural network

See https://doi.org/10.1063/1.5130102

However the flexibility of the **recurrent neural net** construction allowed us to overcome these difficulties by inventing the new network which combine both steps in one **end-to-end TrackNETv2 with the regression part** of four neurons, two of which **predict the point of the center of ellipse on the next coordinate plane**, where to search for track-candidate continuation and another two – **define the semiaxis of that ellipse.**

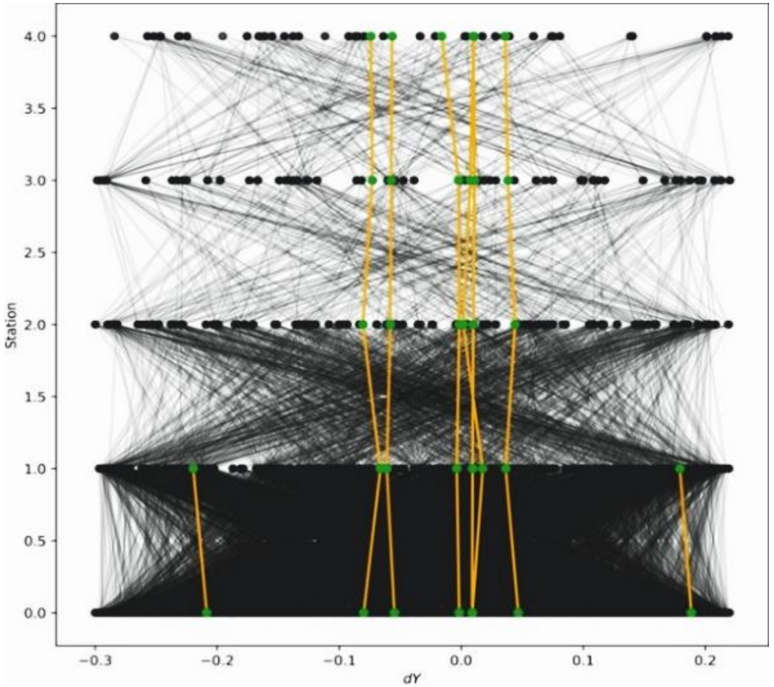
It gives us the opportunity to train a single end-to-end model using only true tracks, which can be extracted from Monte-Carlo simulation.
So we got the neural network performing track following like Kalman filter, although without its track fitting part



2. Global tracking approach

Global recognition of tracks among the noises is carried out at once over the entire picture of the event.

The GraphNet program is based on the use of graph neural networks for tracking. An event is represented as a graph with hits as nodes, and then this **graph is inverted into a linear digraph, when the edges are represented by nodes and the nodes of the original graph are represented by edges**. In this case, information about the curvature of track segments is embedded in the edges of the graph, which simplifies the recognition of tracks in



Graphical representation of the C + C, 4 GeV event of the BM@N experiment. Black nodes and edges correspond to fakes, green nodes and yellow edges - found tracks

See http://ceur-ws.org/Vol-2507/280-284-paper-50.pdf

- During training, the network receives as input a reverse digraph with labels of true edges - segments of real tracks.
- The already trained neural network RDGraphNet (Reversed Directed Graph Neural Network,), as a result, connects each edge with the value x ϵ [0,1] at the output.
- True track edges are those edges for which x is greater than some given threshold (> 0.5).

Two metrics were used to measure the quality of the both local and global models:

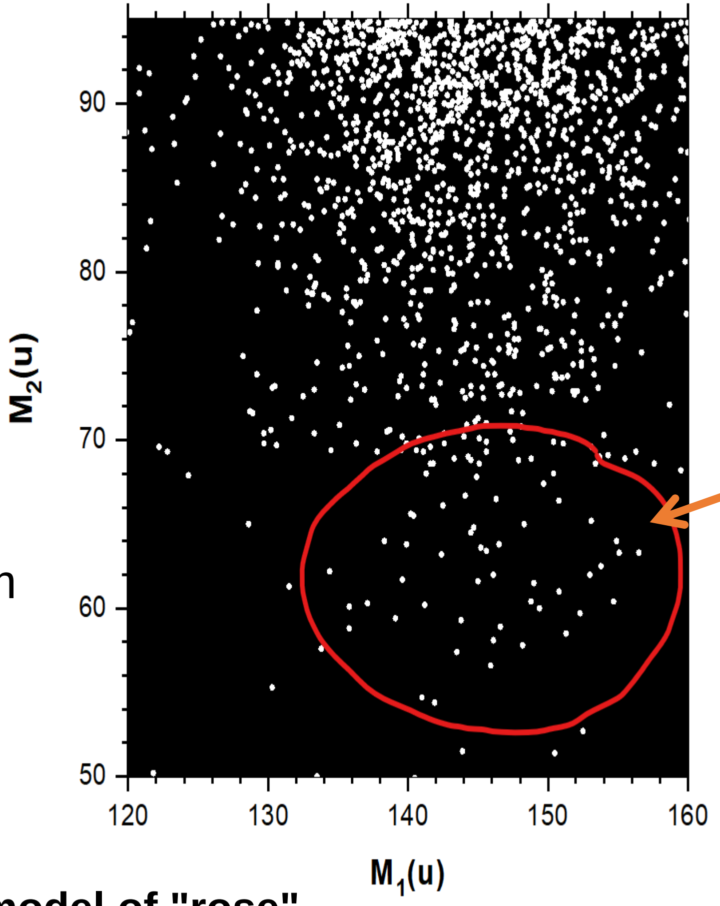
• **Recall:**, where N_{rec}^{true}/N_{MC}– the number of correctly reconstructed true tracks and N_{MC} total number of all modelled real tracks (Monte-Carlo).

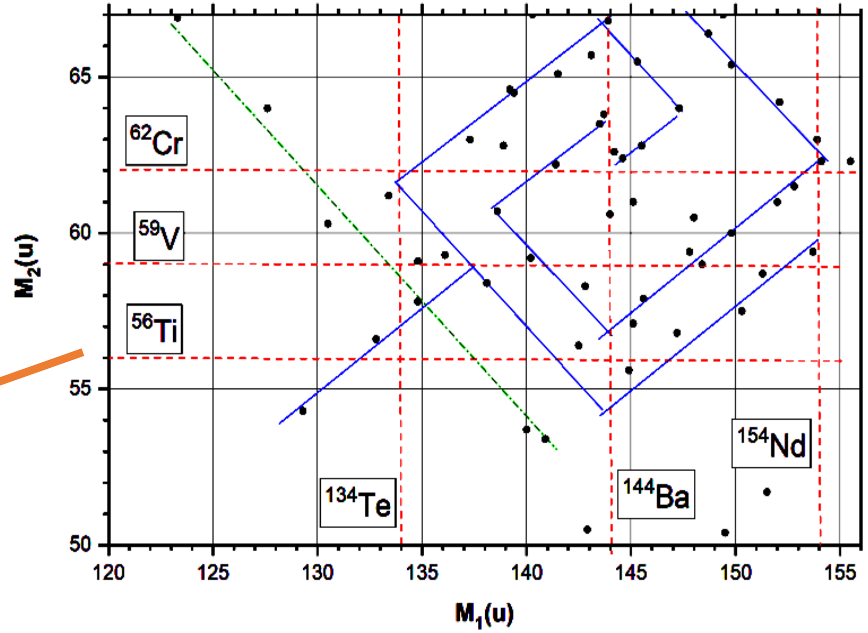
• **Precision:** N_{rec}^{true}/N_{rec}, where N_{rec} – total number of reconstructed tracks (including fake tracks)..

ML in other physical applications -1

1. Data from Nuclear reactions: Californium ^{252}Cf fission analysis

Mass-mass correlation plot of Californium spontaneous fission provided by the team of prof. Yu. Pyatkov





Note: the red dashed lines mark the masses of magic nuclei

Task: prove that this “Nuclear rose" is not a random set of points

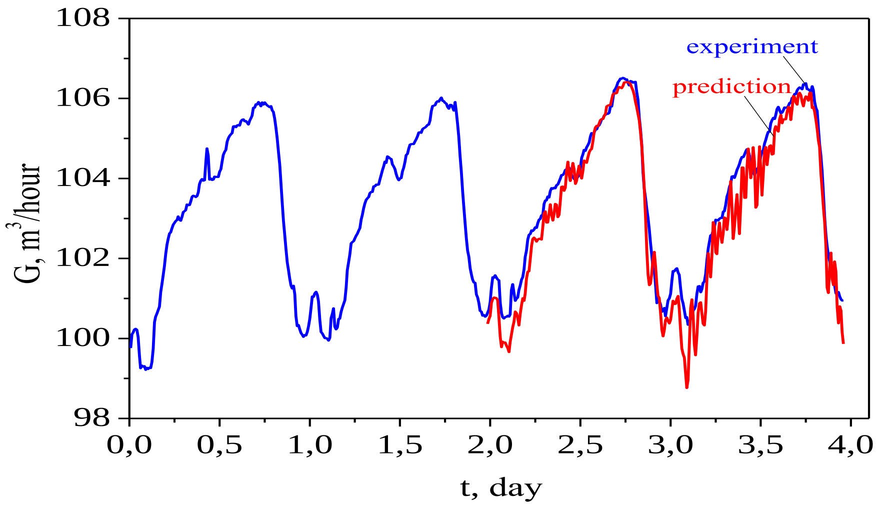
Solution:

- **develop a numerical model of "rose"**
- **create a dataset for training a convolutional ANN used as a neural-classifier**
- **train a convolutional ANN-classifier to distinguish a set of noise points from the "rose" and to make sure that the probability of the wrong prediction of the “rose” among a set of noise points is negligible**

ML in other physical applications -2

3. Data from IBR-2 neutron pulced reactor and neutron activation analysis

The time series is to be predicted for the operational control of the reactor work



Liquid sodium flow rate through the core IBR-2M

Task: predict the liquid sodium flow rate through the core of the IBR–2M during a reactor operation

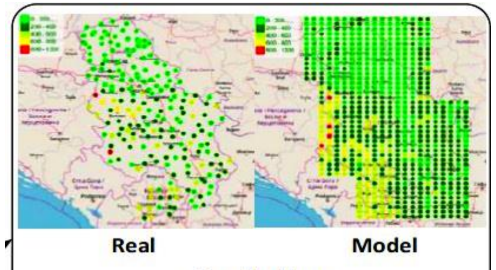
Solution: not by Granger Causality Analysis as in Martina Chvosteková talk, but by applying <u>nonlinear autoregressive neural network (NAR)</u> with the lag n-44 for training and switching on feedback for predicting.

International Program for the monitoring and evaluation of effects of air pollutants on vegetation (ICP Vegetation).

Mosses are ideal absorbers of polluted air. Volunteers collect mosses in some places, send them to JINR for the neutron activation analysis to find out what kind of harmful substances pollute the air in a given area.

Task: extrapolate this data from discrete points of moss collecting on the surrounding area for each of the pollutant taking into account the specifics of the environment, such as wind and gutters directions

Solution: Data Management System (DMS) is developed at the JINR cloud platform providing its participants with modern unified system of collecting, analyzing and processing of biological monitoring data. DMS allows organizing flexible data collection in respect to heterogeneity of raw data, automatic verification of imported data, calculate its statistic parameters, geo-indexes and so on. Now more than 6000 sampling sites from 47 regions of different countries are presented at the DMS



Prediction

Beyond of the field of physics

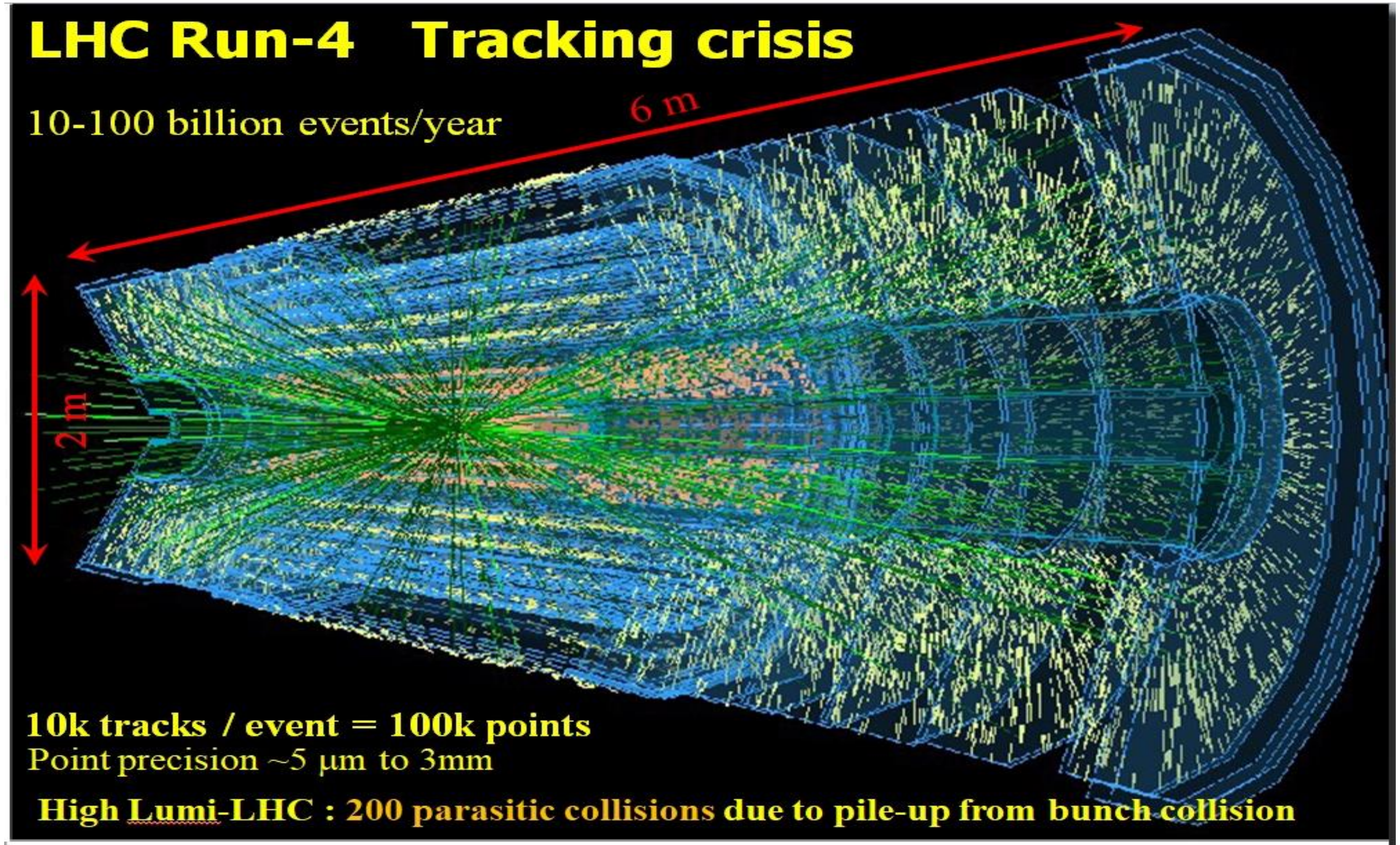
- The era of Big Data has now embraced all areas of science and applications: biology, economics, medicine, etc. So the general demand for new data processing techniques such as deep learning has inevitably become urgent in these areas as well.
- However, the almost fatal difference between these areas from physics lies in the absence of a replaced physical "standard model" that allows generating training samples of any desired length for reliable training of the developed deep neural networks.
- The lack of labeled data for training becomes one of the main problems. In addition to the lack of theoretical prerequisites, the limited set of training data is explained by both the high cost of the experiments themselves and the cost of the work of experts who mark up the data obtained.

Let us take as an example the problem of some **CNN agriculture applications**

Disease detection on the plant leaves by deep learning

One can see our article **DOI: 10.1007/978-3-030-01328-8_16** for details, where the web platform http://pdd.jinr.ru is described, which allows users to use a smartphone to transmit photos and a brief description of sick leaves and get a response with a diagnosis of the disease and the recommendation for treatment.
Although there are many mobile applications for plants disease detection, as for example, famous plantix.net, but none of them allowed us to get a dataset suitable for use for us. We had to create our own, to work with 1556 images of leaves, sick and healthy, obtained in real conditions for 44 different diseases of 5 classes of plants: grapes, wheat, corn, cotton, cucumbers. To overcome the lack of labeled data obstacle we use a **Siamese CNN with a special triplet loss function and a perceptron, as a classifier in the output layer, that shows 98% of accuracy**

What is the next in HEP



Particle track reconstruction in dense environments such as the detectors of the High Luminosity Large Hadron Collider (HL-LHC) and of MPD NICA is a challenging pattern recognition problem.

For those who are interested in machine learning:
On the site http://gososkov.ru/uni-dubna/machine%20learning/
You can download my lectures on machine learning and textbook S. Nikolenko in Russian

Thank you for attention!