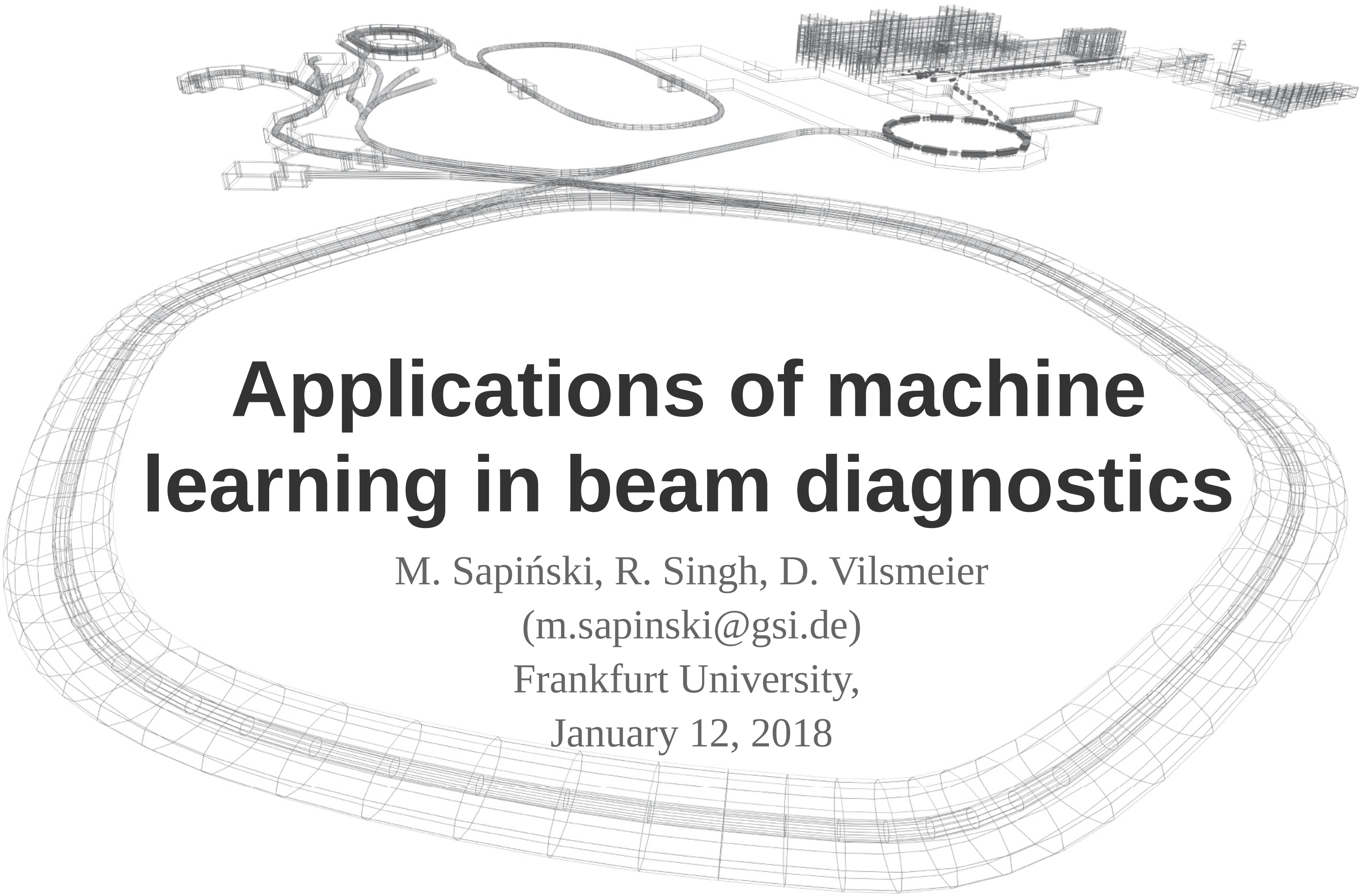


Applications of machine learning in beam diagnostics

M. Sapiński, R. Singh, D. Vilsmeier
(m.sapinski@gsi.de)

Frankfurt University,
January 12, 2018

Outline

- Introduction: what is Machine Learning?
- Some famous examples.
- Artificial Neural Networks.
- Theoretical background.
- Example 1: identification of quench-provoking loss patterns at LHC.
- Example 2: correction to measured beam profile distortion in Ionization Profile Monitor.
- Conclusions.

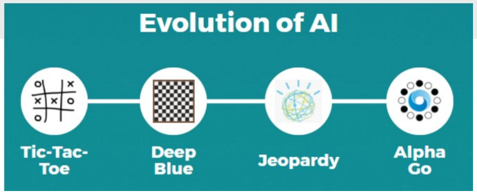
Remarks and Disclaimer

- Beam Diagnostics takes care of beam parameters measurements, for example: beam position, beam current, longitudinal and transverse profile, beam loss, tune, chromaticity, etc.
- Machine Learning techniques are also used in other aspects of accelerators, mainly control systems, machine tuning – not discussed here.
- Keep in mind that I am ML enthusiast, but not trained Machine Learning professional.

What is Machine Learning?

- Algorithms which can learn and make predictions on data, without explicit programming.
- The term by Arthur Samuel (IBM) in 1959.
- Machine learning is closely related to computational statistics and to mathematical optimization.
- Data mining is a sub-field of Machine Learning known as unsupervised learning.
- Expert systems – are made of digitized/encoded expert knowledge. They are not Machine Learning algorithms. Still useful is there is little data available for training. Mixed systems are also available.

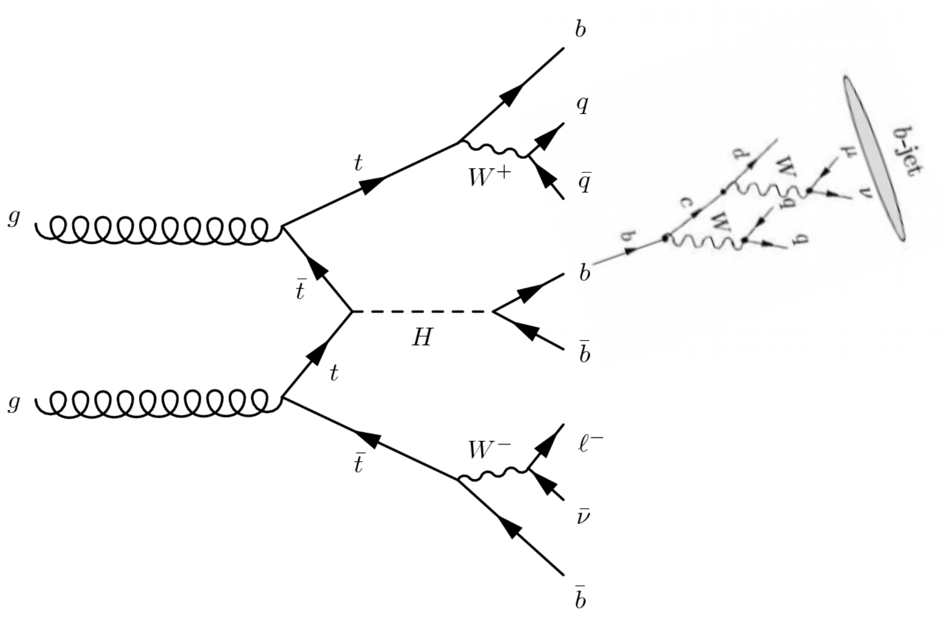
Examples of ML-based projects (I)

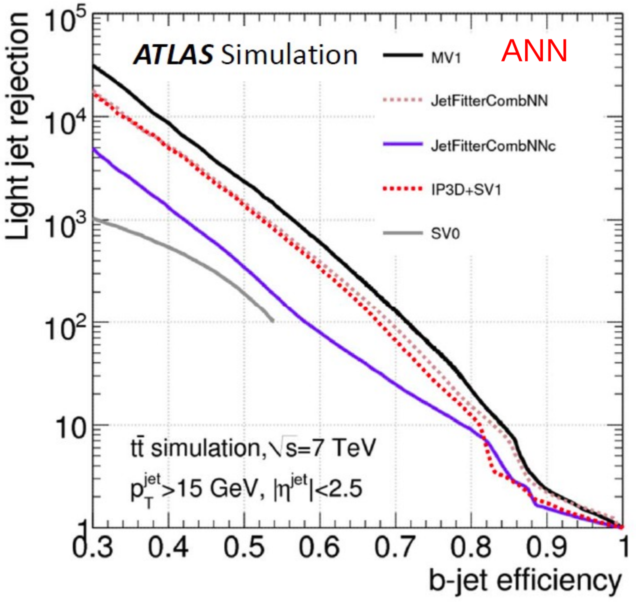


- AlphaGo :
  - Go is difficult for algorithms because of number of configurations (>$2\text{x}10^{170}$,chess only ~$5\text{x}10^{52}$), atoms in the Universe ~ 10^{80}.
  - The program uses Artificial Neural Network for learning and Monte Carlo Tree Search for decide about next move.
  - 1 year learning time, 183 MWh energy, excessive data sample – not the way human learns, but:
    - AlphaGo won against the highest-qualified humans.
    - It has exhibited creative skills making moves seldom done by humans.

Examples of ML-based projects (II)

- Neural Networks are used in physics analyses since ~1988.
- They were for instance used to reject background in Higgs boson search – but published analysis does not use ML.
- b-tagging:



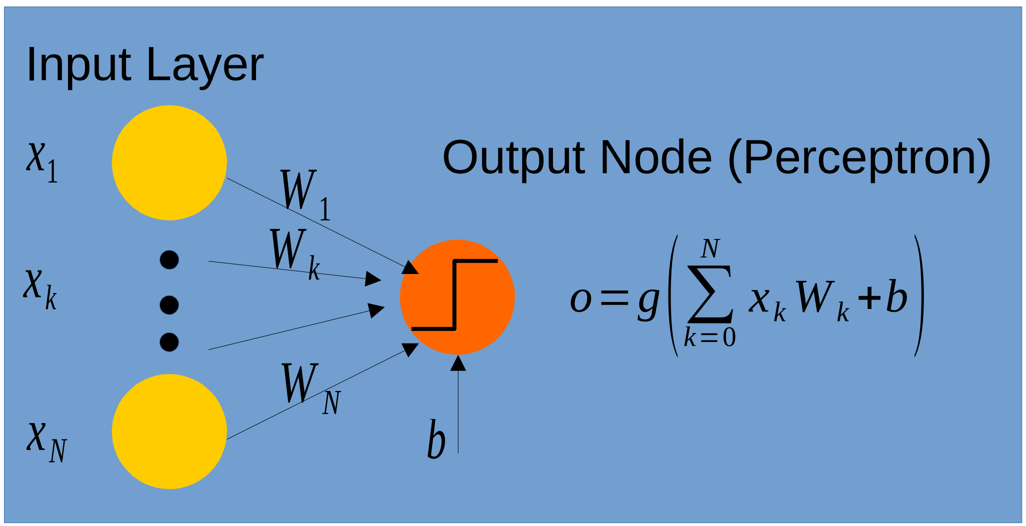


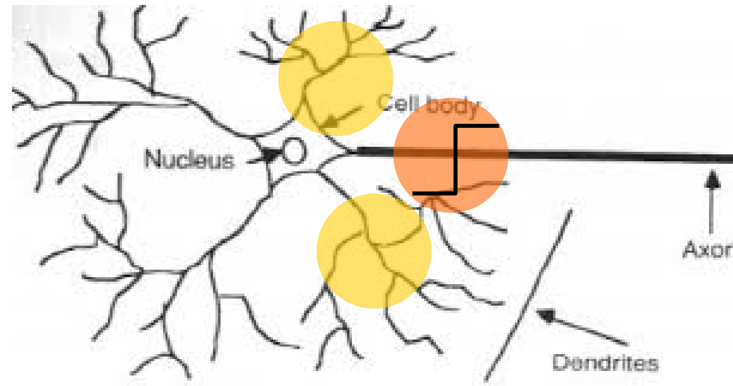
the scope of this presentation is:

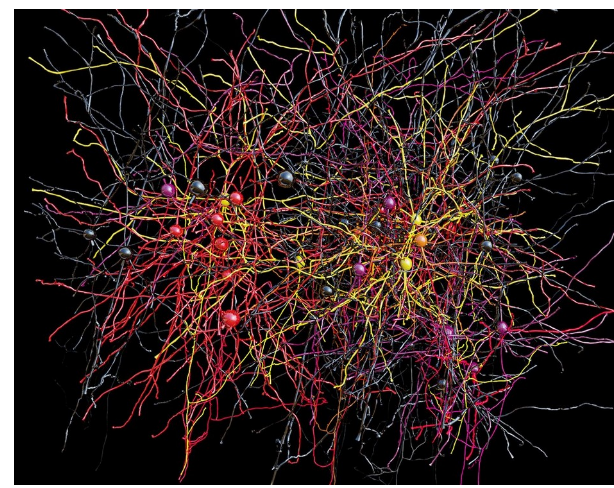
- show examples how Machine Learning techniques, mainly artificial neural networks (ANN), can be useful to solve everyday problems of accelerator physicist in domain of beam instrumentation.

Artificial Neural Network

- Biologically inspired → Brain cells -> neurons, computation via connections and thus Networks
- The basic node of ANNs is “Perceptron”





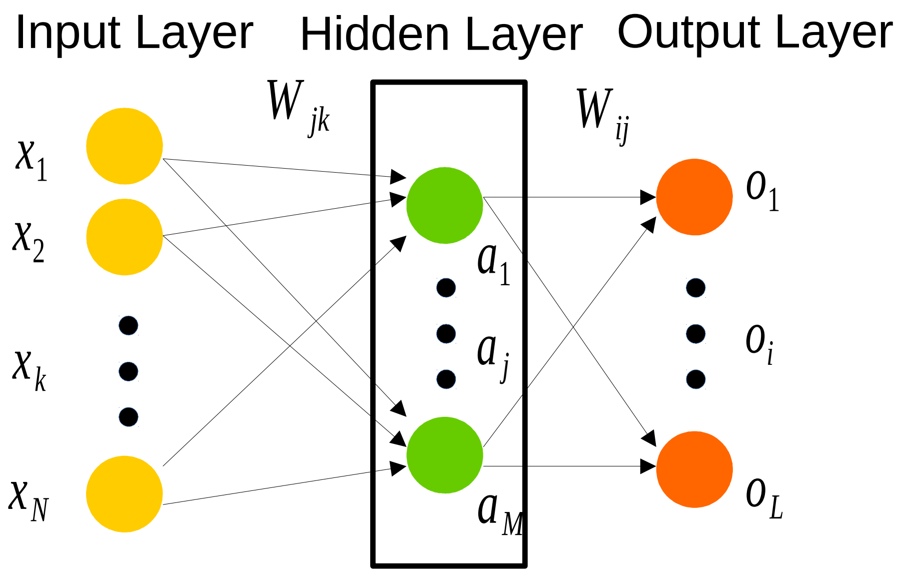


<u>Perceptron parameters:</u>

- Weights from the inputs (X) and bias (b)
- g is the activation function, a step-like function with a threshold

[https://www.wired.com/2016/03/took-neuroscientists-ten-years-map-tiny-slice-brain]

Hidden layers



Multi-layer Perceptron

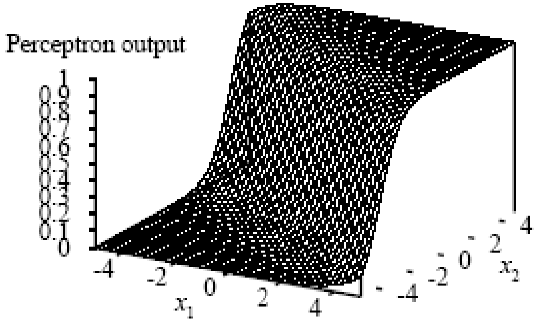
Each hidden layer and output layer node is a perceptron

$$o_i = g\left(\sum_{j=0}^{M} W_{ij}\left(g\left(\sum_{k=0}^{N} x_k W_{jk} + b_j\right)\right) + b_i\right)$$

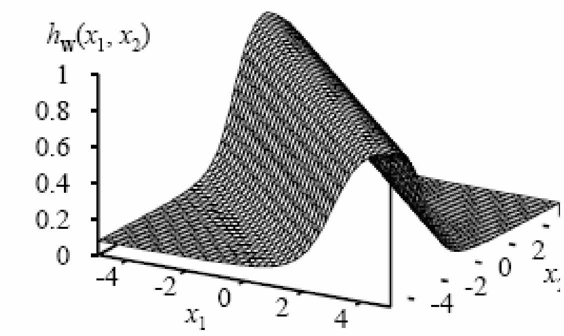
Adding “hidden” layer(s) allow non-linear target functions to be represented

Multi-layer perceptron (MLP)

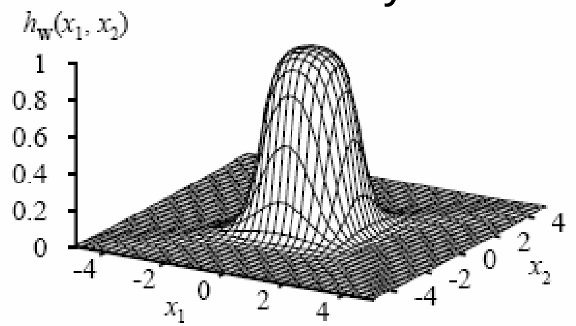
Perceptron: No hidden layer



One hidden layer



Two hidden layers



- Carla P Gomes, Lecture Notes CS 4700: Foundations of Artificial Intelligence

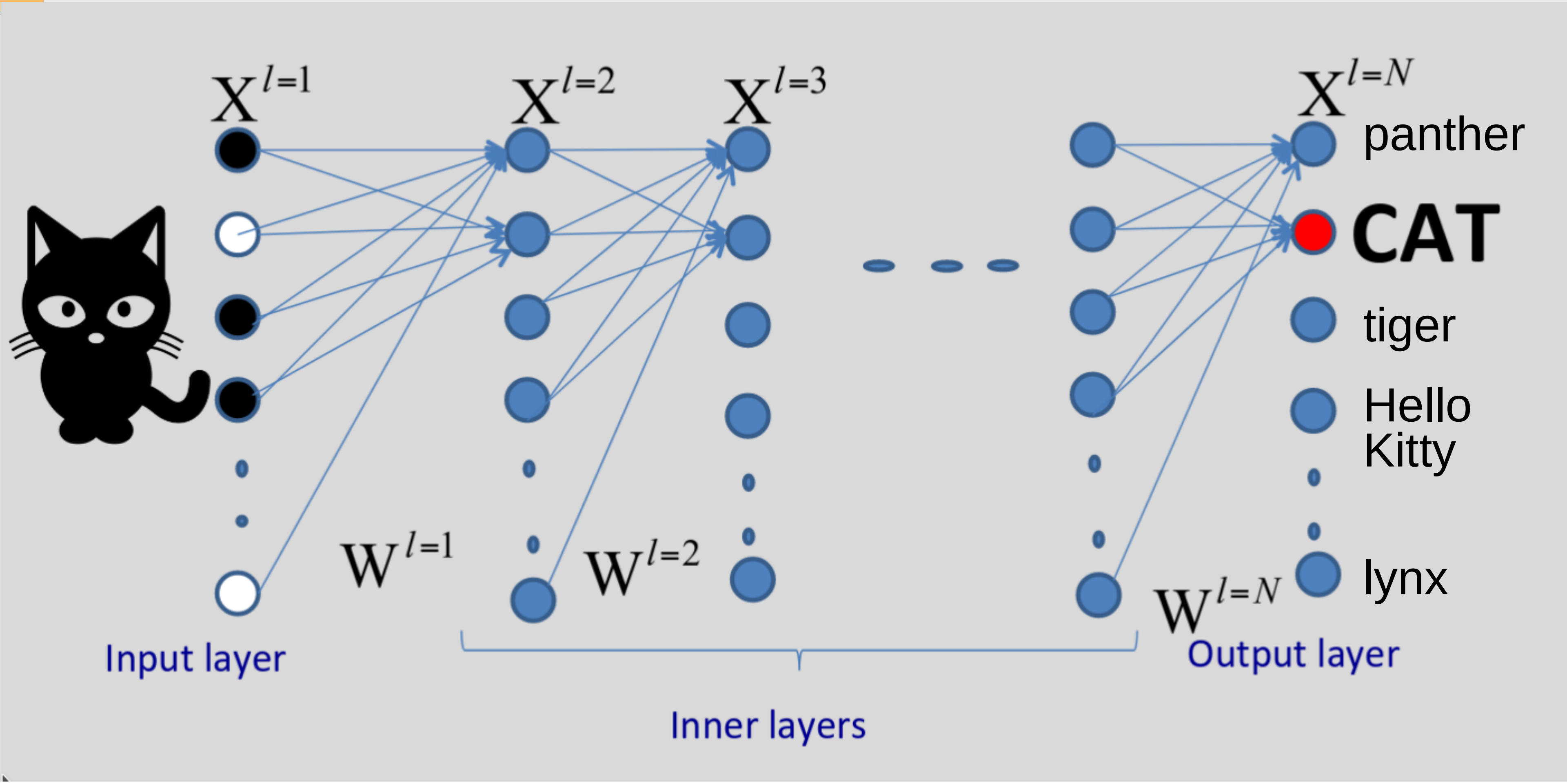
Universal approximation theorem:

- Every bounded continuous “target” function can be approximated with arbitrarily small error, by network with single hidden layer [Cybenko 1989; Hornik et al. 1989]

If we have any unknown function, $y=f(x)$,it can be approximated by:

$$o_i=g\left(\sum_{j=0}^{M} W_{ij}\left(g\left(\sum_{k=0}^{N} x_k W_{jk}+b_j\right)\right)+b_i\right)$$

Multi-layer perceptron (MLP)



How to design MLP topology for a given problem?
How to find the weights? (train network)

MLP Network design

- For feed-forward MPL.

From: https://www.solver.com/training-artificial-neural-network-intro

- *There is no best answer to the layout of the network for any particular application. There are general rules:*
  - As the complexity between input and output increases, the number of the perceptrons in the hidden layer should also increase.
  - If the process being modeled is separable into multiple stages, then additional hidden layer(s) may be required. Otherwise additional layers may simply enable memorization of the training set, and not a general solution effective with other data.
  - The amount of training data sets an upper bound for the number of perceptrons in the hidden layer(s).
    If you use too many perceptrons the training set will be memorized.
    ->generalization of the data will not occur, making the network useless on new data sets.

MLP Network training

- Some algorithms known since 40's (Gauss Newton or Levenberg-Marquardt).
- Backpropagation with Gradient Descent developed in 70's – speeds up in ANN training – it triggered a wave of interest in ANN applications – still most popular.
- How it works:
  - Activation function g must be differentiable, eg. sigmoid or tanh.
  - Initial weights chosen randomly.

$$o_i = g\left(\underbrace{\sum_{j=0}^{M} W_{ij}\left(g\left(\sum_{k=0}^{N} x_k W_{jk} + b_j\right)\right) + b_i}_{inp_i}\right)$$

- For training record (or a batch of records) a cost function (or loss or error) is calculated, for instance mean squared error:
(y-desired output, o-actual output)

$$E = \sum_{i=0}^{L} \left(y_i - o_i\right)^2$$

- The cost function gradient is calculated for each layer:

$$\frac{\delta E}{\delta W_{ij}^{1}} = a_j Err_i g'(inp_i) \qquad \frac{\delta E}{\delta W_{jk}^{2}} = x_k g'(inp_j) \sum_{j=0}^{M} W_{ij} Err_i g'(inp_i)$$

- New weights are calculated:
- Repeat for new record
(but you can use the same record later again)

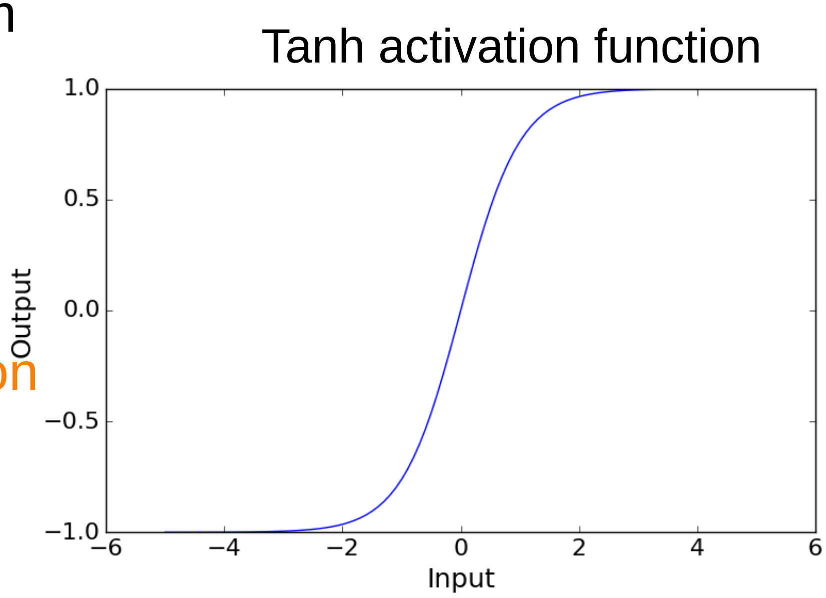
$$\boxed{W(t+1) = W(t) + \alpha \frac{\delta E}{\delta W}}$$

α-learning rate

Conditioning inputs and initial weights

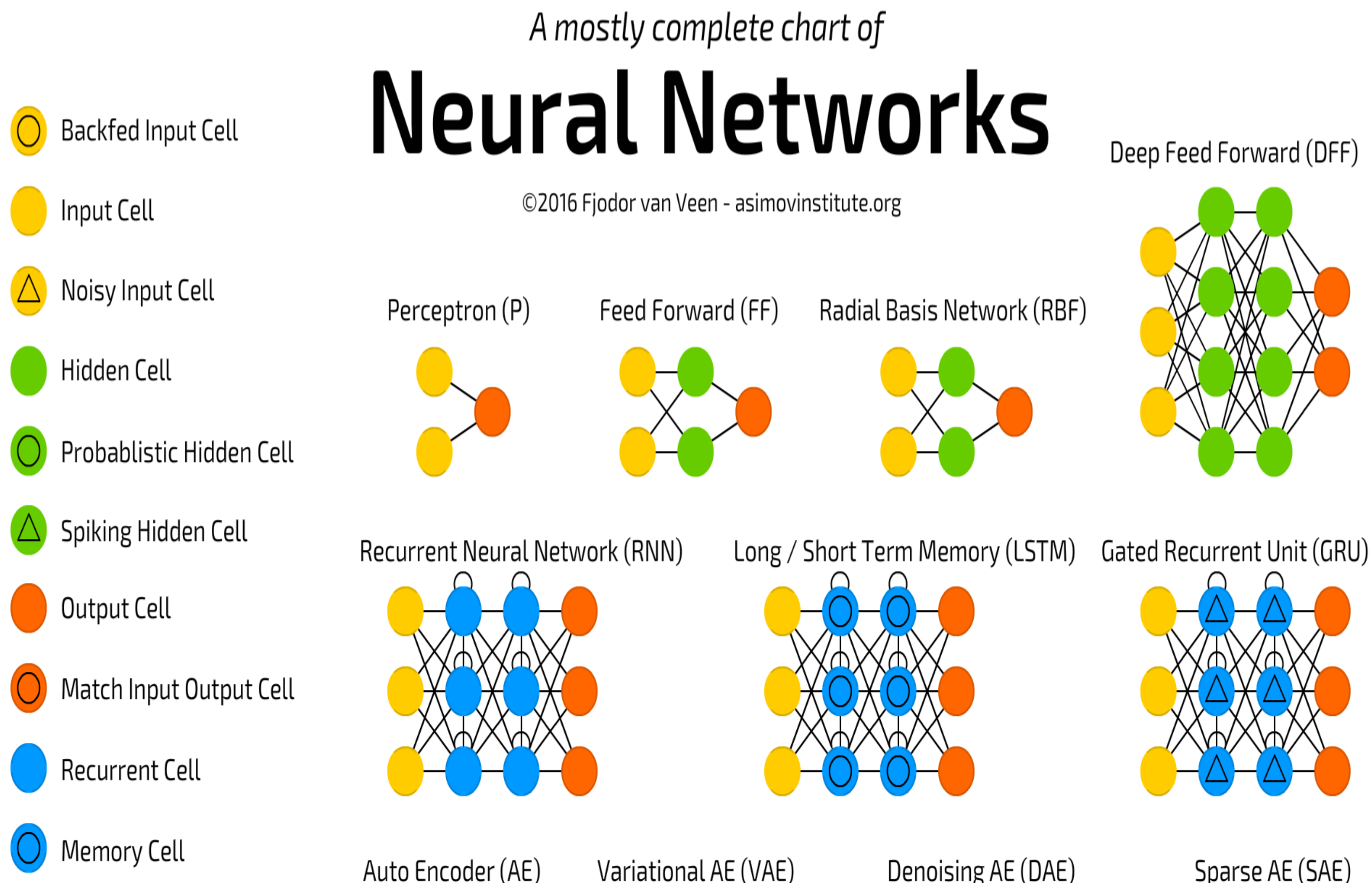
- Weights initialization: Generate random initial weights [-1,1] and divide each of by the square root of the number of units in the larger layer.

- Inputs and targets to be normalized according to the used activation function (tanh: -1..1, sigmoid: 0...1) , else some perceptrons will remain saturated (difficulty in learning).



- Rules of thumb: Start with two hidden layers with number of hidden units equal to (Input_num + Target_num)/2, avoid overfitting by regularization.
- If simple MLP is not good enough for the application, look further into literature!

Neural network Zoo



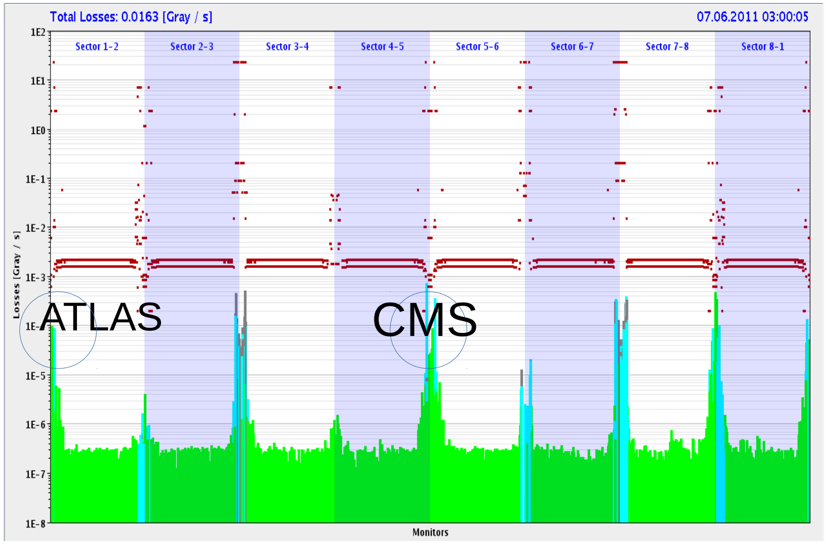
… 20 more

Source: Fjodor Van Veen, Asimov Institute, Utrecht

Example 1: Pattern recognition in BLM signals

Beam Loss Monitors (BLMs) at LHC:

- Most high-power accelerators are equipped with BLM systems.
- LHC beam has energy of about 360 MJ per beam, equivalent to about 300 passenger cars on a motorway.
- Uncontrolled loss of even fraction of such a beam can damage equipment or quench a magnet.
- Therefore about 4000 BLMs are installed around LHC, ready to dump the beam within ~200 µs.

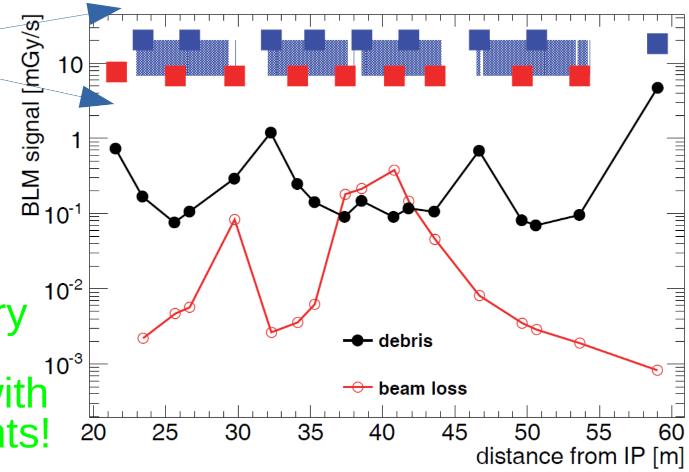


Example 1:
LHC interaction points

- In order to focus the beams in the interaction point (experiment) special high-gradient quadrupoles are installed – called **triplets**.
- Beam-beam collisions produce interesting physics results and debris, which leak to triplets
- Due to that triplet magnet are constantly “heated” to about 30% of quench limit (~3 mW/cc).
- Only small variation of BLM signal corresponds to quench-provoking beam loss. What to do?

ATLAS /CMS

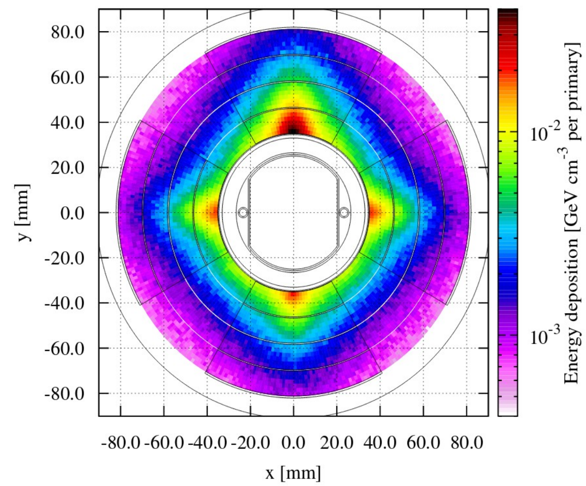
debris



FLUKA simulations: for debris very good agreement with measurements!

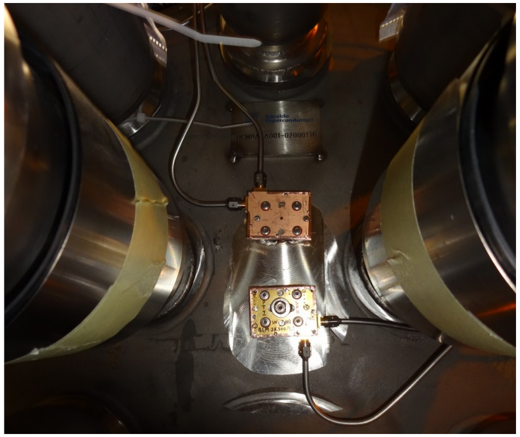
So:

Energy inside magnets shou also be well estimated!



Example 1:
How to recognize beam loss

- First idea: install BLM monitors inside the magnets, close to the coils.
  - Advantage: closer to the coil – measurement corresponding better to the real coil heating
  - Disadvantages: difficult location, small space, no service possible, liquid helium environment, high integrated dose, technical risk due to additional structures inside magnet etc…
  - R&D and test installation done with silicon and diamond detectors, not very promising!



- Test Artificial Neural Network patter recognition capacity!

Example 1:
Some code

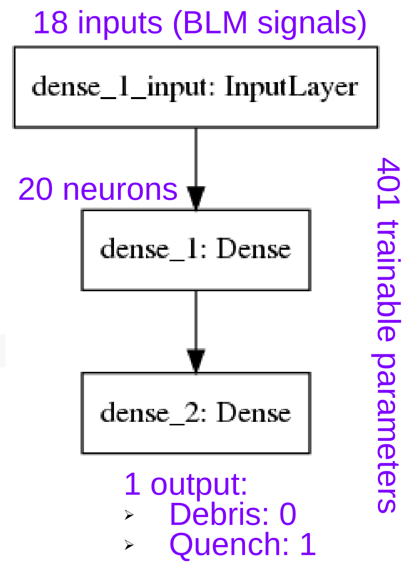
- Python
- Google tensorflow library with keras interface:

  >pip install tensorflow

  >pip install keras
- Create ANN:

```
from keras.models import Sequential,Model
from keras.layers import Input, Dense, Activation
import matplotlib.pyplot as plt
import numpy as np

model = Sequential()
model.add(Dense(20, input_dim=18, activation='tanh'))
model.add(Dense(1, init='uniform', activation='sigmoid'))
model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['accuracy'])
```



- Prepare data:
  - Debris: 100k events with independent random variation of each BLM signal by 10%
  - Loss: 100k events with independent random variation of each BLM signal by 50% MIXED with debris to a quench level.

- Train: model.fit(traindata,trainlabel,nb_epoch=70)
- Run on new data: out_loss=model.predict(testdata)

that's it! 10 – 20 lines of code!

Example 1:
Pattern recognition in BLM signals

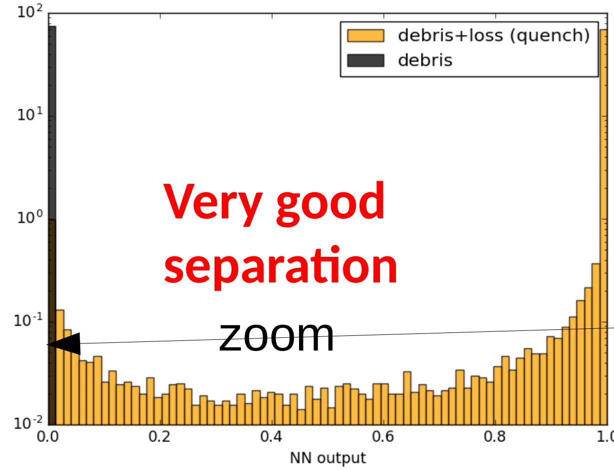
Result:

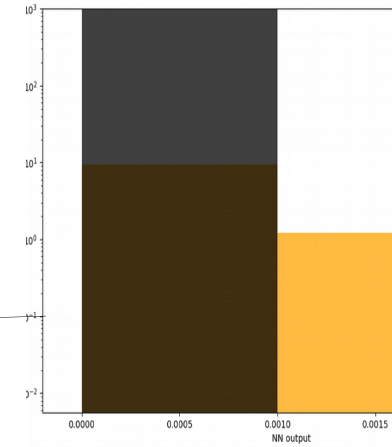
Overlap: about **0.5%** of quench-provoking losses undetected.

Compare it to standard method, for instance
Mean Square Error between expected debris signal (D_{exp}) and signals at quench.

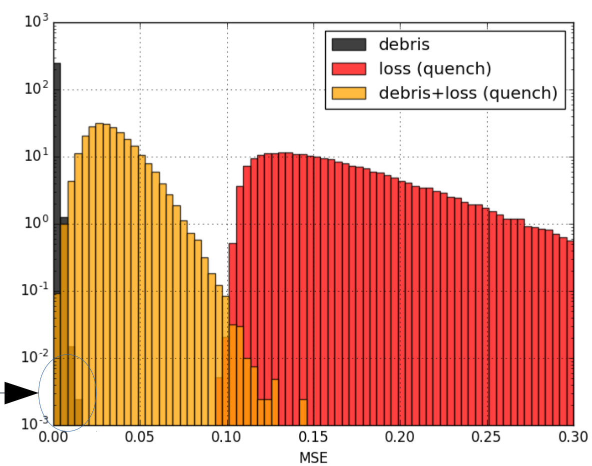
$$MSE=\Sigma(D_{exp,i} - S_i)^2$$

To give more chances to classical signal we limit ourselves to 6 most sensitive BLMs. Otherwise it is the same data.
Overlap: about **2%** of losses undetected.





0.001



Example 1:
Pattern recognition in BLM signals

Lessons learned:

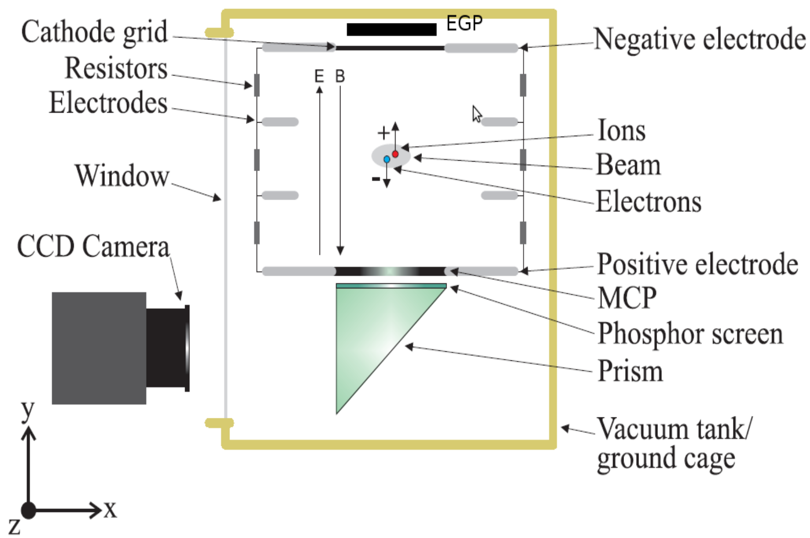
- It took < 1 day to perform this analysis: it is easy!
- Without further optimization the results are better than (simplistic) “classical” approach.

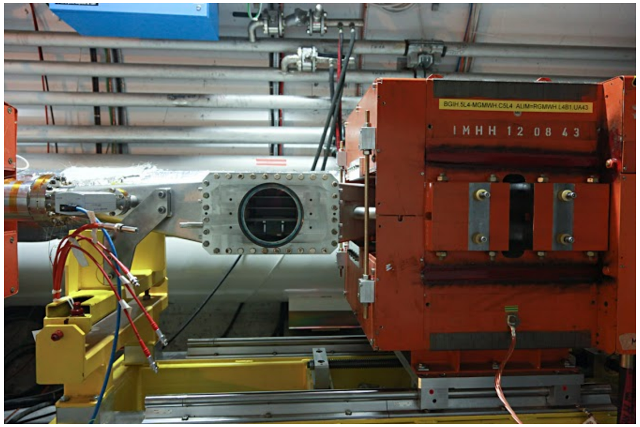
- However this solution was finally not chosen, because people do not like “black boxes”...

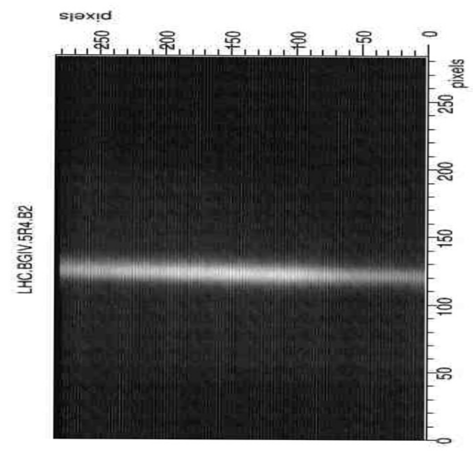
Example 2: Correction of IPM signal distortion due to beam space-charge

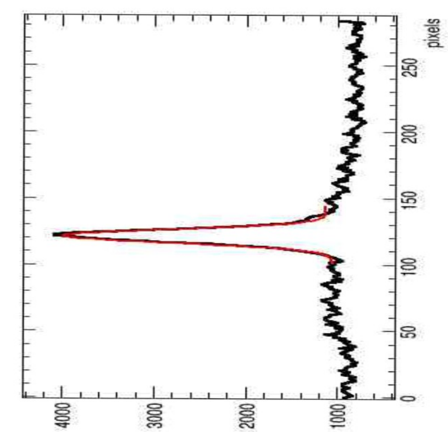
<u>Ionization Profile Monitor (IPM):</u>

- Measures transverse profile of particle beam.
- Rest gas (pressure 10^{-8} mbar) is ionized by the beam.
- Electric field is used to transport electrons/ions to a detector.
- If electrons are used – additional magnetic field is usually applied to confine their movement.



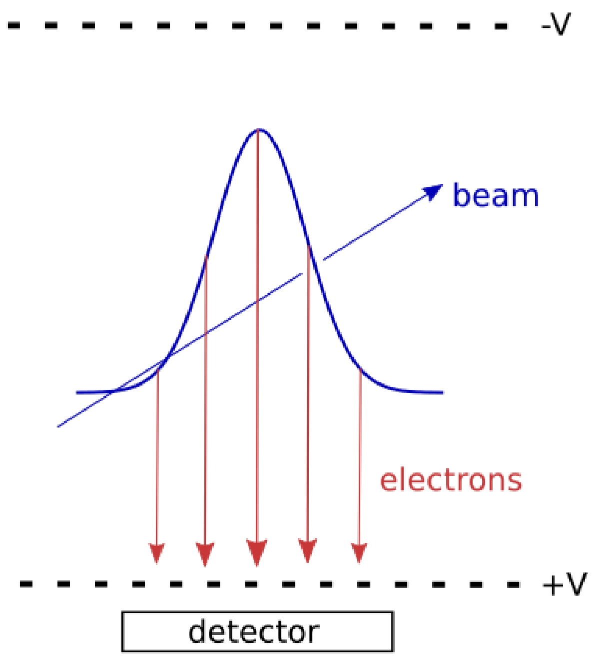




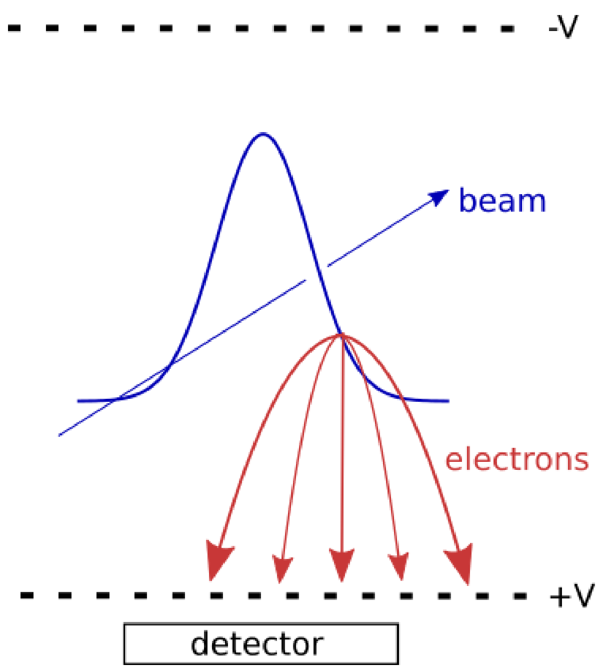


Example 2:
Profile distortion in IPM

Ideal case
Particles are moving on straight lines towards the detector



Real case
Particle trajectories are influenced by initial momenta and by the interaction with the beam field

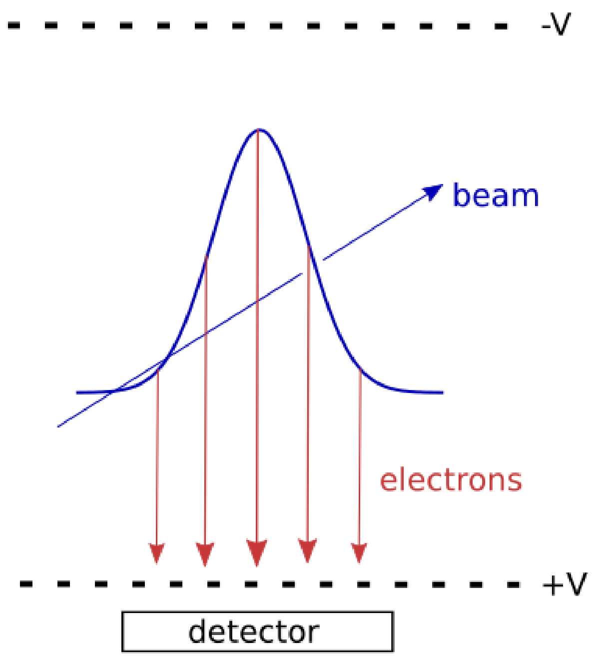


… instrumental effects such as camera tilt, optical point-spread-functions, point-spread functions due to optical system and multi-channel plate granularity etc, etc… come on top!

Example 2:
Profile distortion in IPM

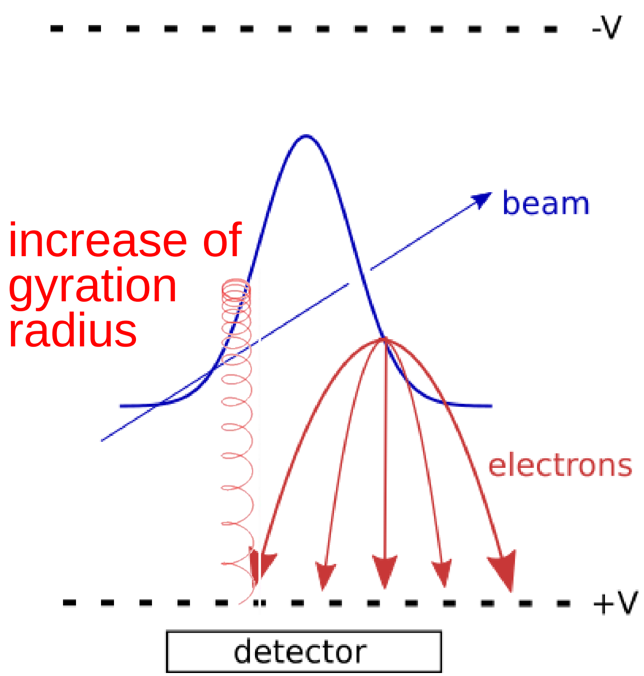
Ideal case

Particles are moving on straight lines towards the detector



Real case

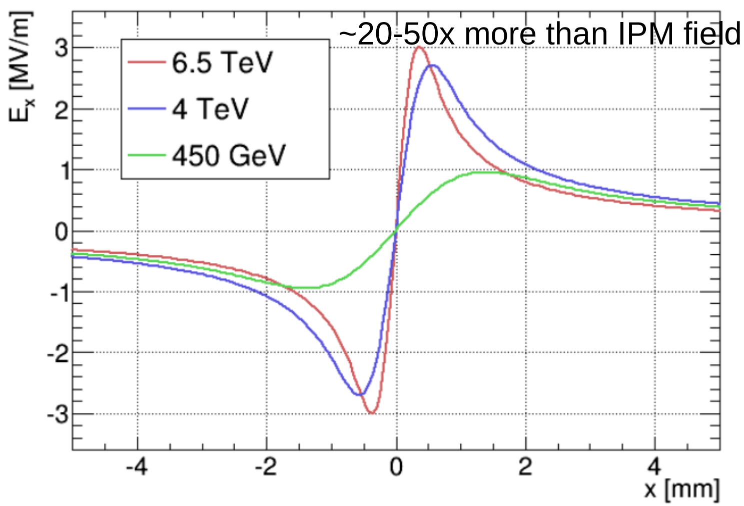
Particle trajectories are influenced by initial momenta and by the interaction with the beam field

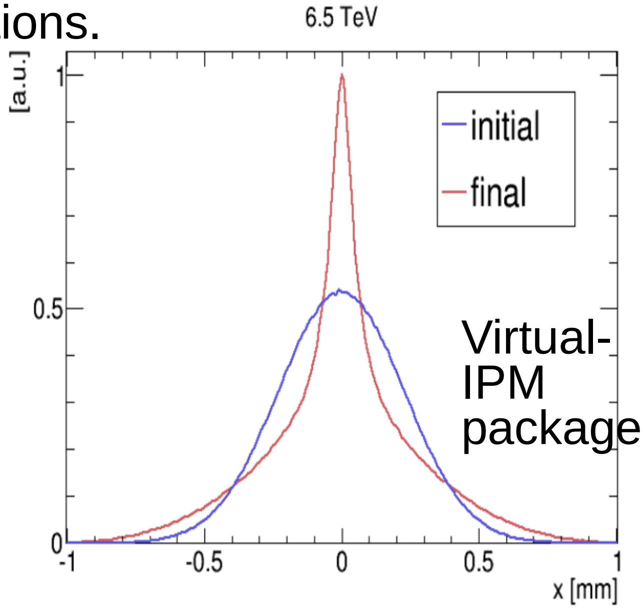


… instrumental effects such as camera tilt, optical point-spread-functions, point-spread functions due to optical system and multi-channel plate granularity etc, etc… come on top!

Example 2:
Profile distortion in IPM - simulation

- Focus on beam field influence.
- Electrons "feel" beam fields (E) and their movement is influenced accordingly resulting in possible displacements.
- This occurs for large beam fields ↔ large charge densities, large beam energies.
- Can be simulated with reasonable assumptions.





Example 2:
IPM profile corrections

- No simple mathematical procedure exists.
- Using higher electric and magnetic fields (expensive, sometimes impractical).
- Electric and magnetic fields : Sieve method (deconvolve with PSF of radius of Gyration).

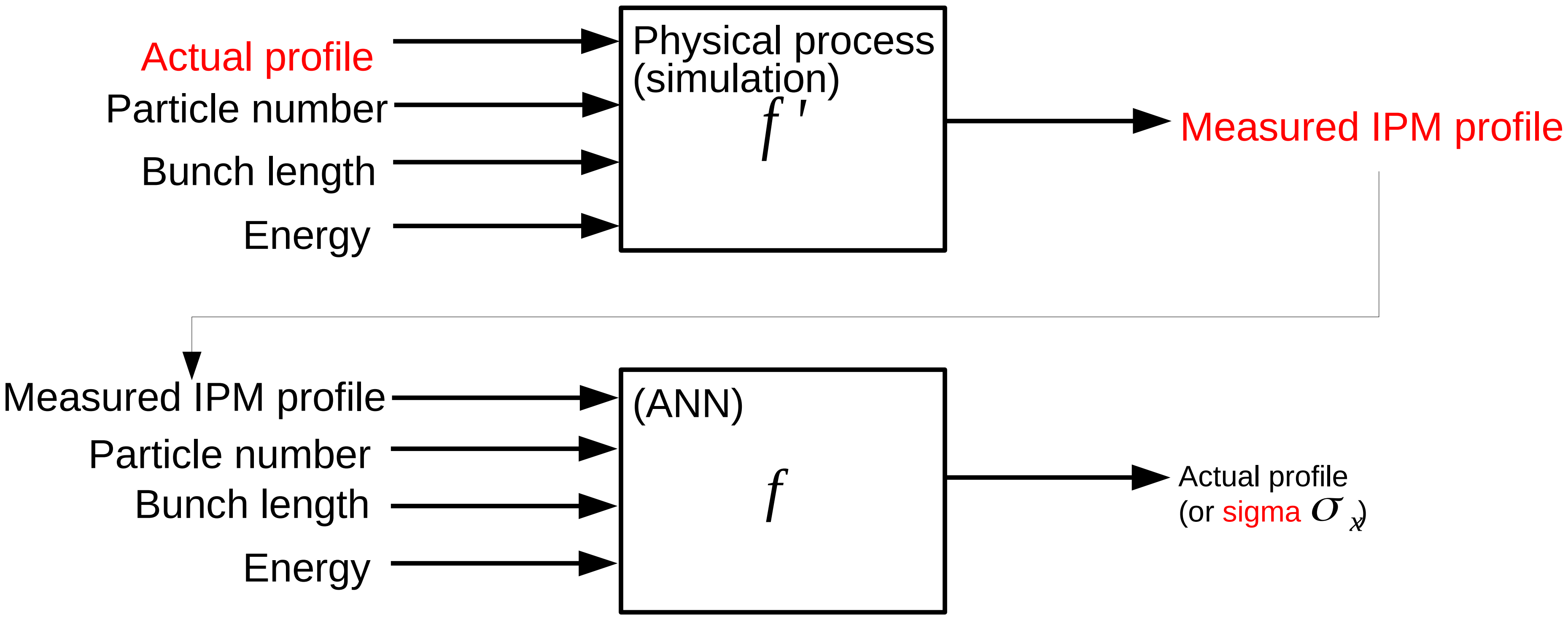
  [Dominik Vilsmeier, Bachelor Thesis, CERN]

- Electric fields only: Several calibration/correction attempts.

  Latest: Assumption on input beam distribution (Generalized Gaussian) and iterative procedure for input reconstruction from distorted profile using the data generated from simulation tool

  [Jan Egberts, PhD Thesis, CEA Saclay]

Example 2:
profile correction using ANN



Training "grid" (375 points):

σ_x 0.29, 0.31, 0.33, 0.35, 0.37 (mm)

σ_y 0.4,0.45,0.5,0.55,0.6 (mm)

N_p 1.1e11, 1.25e11, 1.40e11, 1.55e11, 1.7e11

σ_l 0.9, 1.05, 1.2 (ns)

Using *tensorflow* and
Matlab NN toolbox

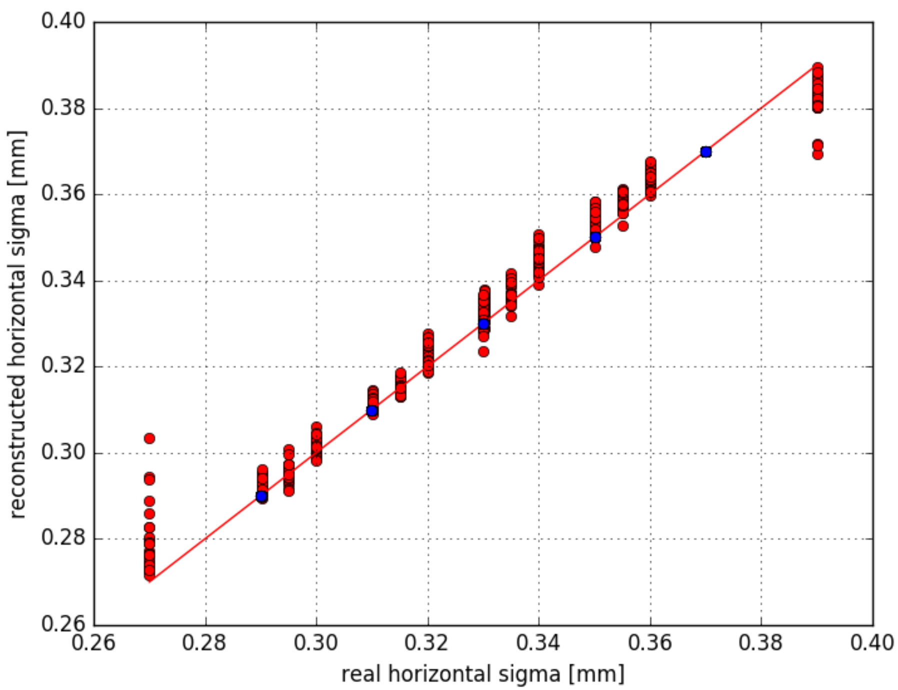
Example 2: Results

Validation “grid” (128 points)

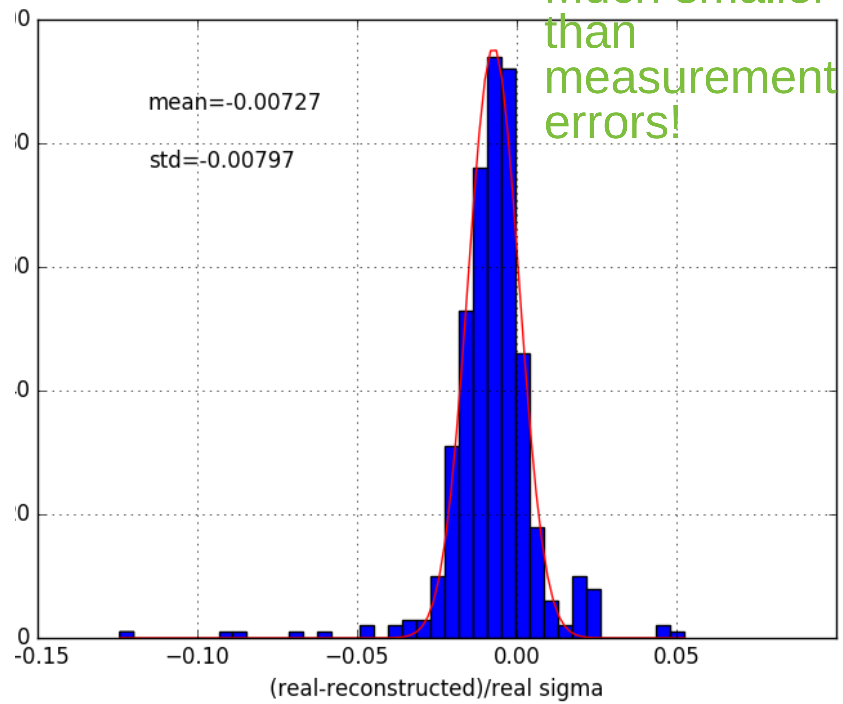
4 validation data sets (inputs and outputs) created:

- 1% off the training grid in each dimension (within in grid)
- 25% off the training grid in each dimension
- 50% off the training grid in each dimension
- 100% off the training grid (the next point outside the grid)

For 12 runs: sigma systematically overestimated by 0.4% with error 0.8%



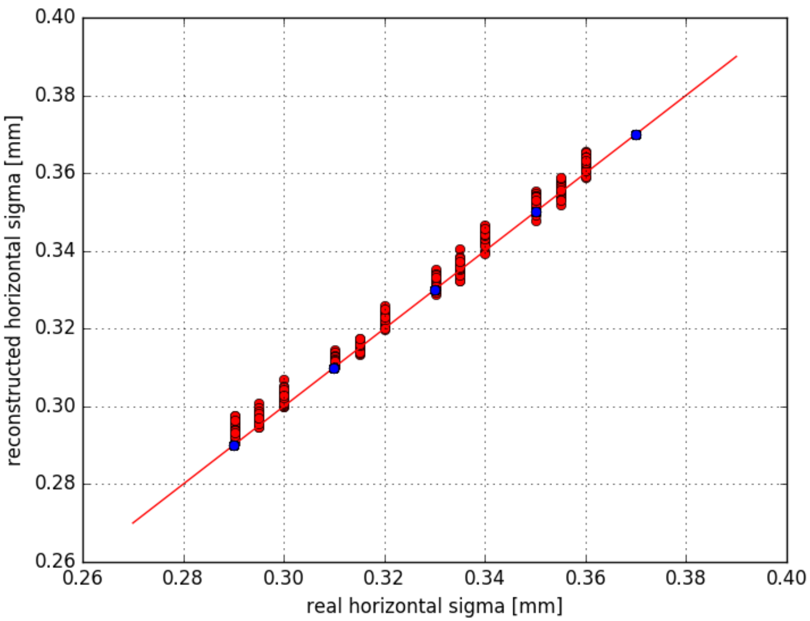
Much smaller than measurement errors!

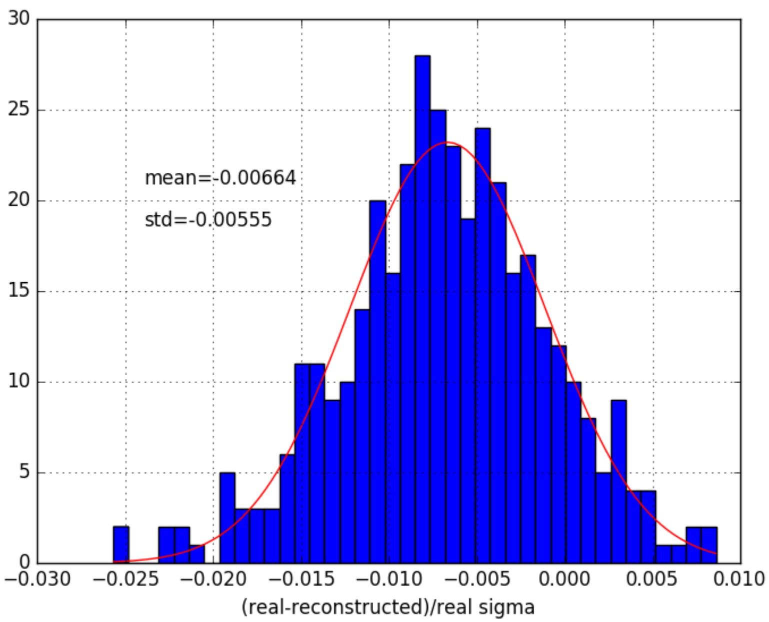


Example 2: Results

Removing the validation sample outside of “training” area

For 12 runs:
sigma systematically overestimated by 0.05% with error 0.7%

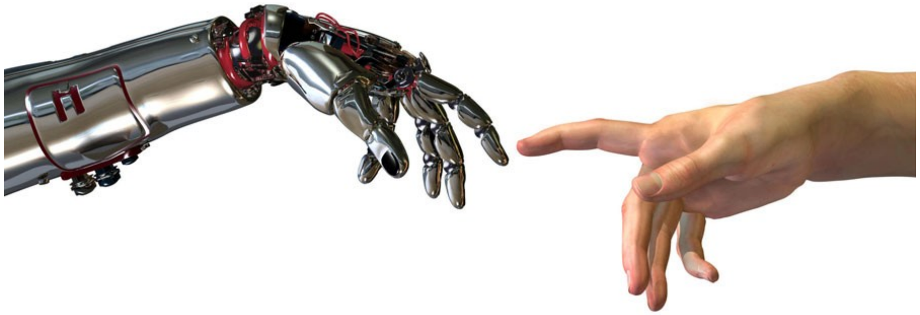




Conclusions

- ML techniques become a standard tool for physicists and engineers.
- They proof to be efficient in solving non-linear multivariate problems with unprecedented accuracy!
- Can save lots of money:

  (CryoBLM project ~1 M€, a set of 1T magnets for IPM ~ 5M€)
- Modern tools (eg. tensorflow+keras) are very easy to use.
- Lot of physicists remain skeptical because “black box” nature of ML and lack of convincing way to estimate errors.
- Personally I think that we should use it but not forget about its limitations.



Further reading and playing

- “How could a Kangaroo climb Everest?” - about minimization algorithms: ftp://ftp.sas.com/pub/neural/kangaroos
- ANN recognizing drawings: https://quickdraw.withgoogle.com
- Music composed by AI: http://www.flow-machines.com/ai-makes-pop-music/
- Unreasonable effectiveness of ANN: http://karpathy.github.io/2015/05/21/rnn-effectiveness/
- E. Musk concerned about AI: https://www.youtube.com/watch?v=0NTb10Au-Ic
- AI algorithms in social media – very interesting: https://www.ted.com/talks/zeynep_tufekci_we_re_building_a_dystopia_just_to_make_people_click_on_ads
- ANN playing with images: https://nerdist.com/why-are-googles-neural-networks-making-these-brain-melting-images
- …