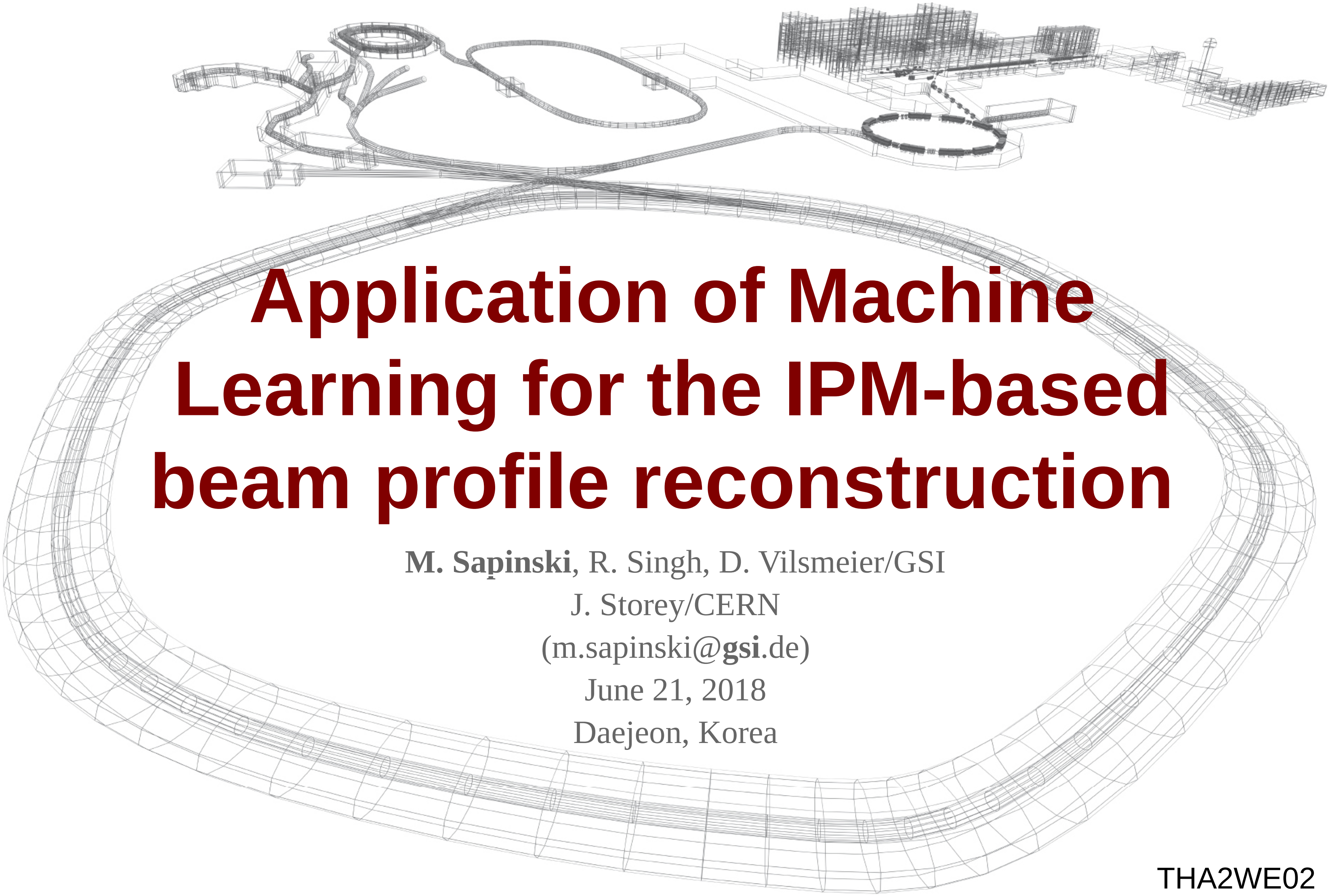


Application of Machine Learning for the IPM-based beam profile reconstruction

M. Sapinski, R. Singh, D. Vilsmeier/GSI
J. Storey/CERN
(m.sapinski@**gsi**.de)

June 21, 2018
Daejeon, Korea

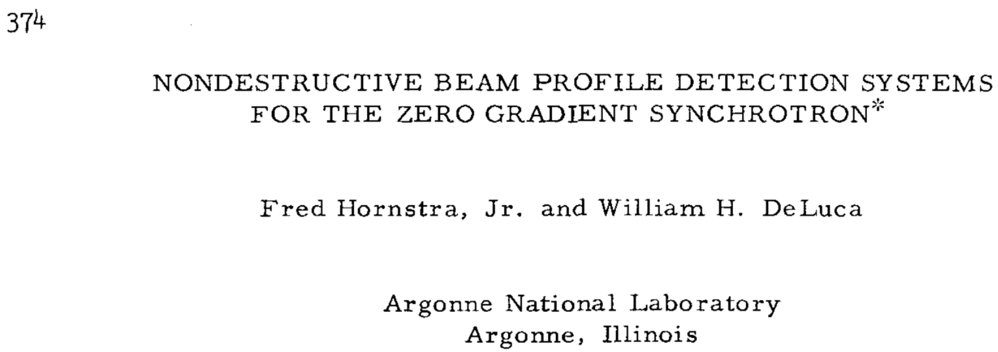
THA2WE02

Outline

- Ionization Profile Monitors
- Profile distortion, previous approaches
- Virtual-IPM program
- High space charge regime: LHC beams
- Understanding profile distortion
- Correction using electron sieve
- Machine learning and Artificial Neural Networks
- Corrections based on Machine Learning
- IPM for micron-size beams using space charge
- Conclusions

Introduction to IPMs

- First constructed in Argonne National Laboratory to measure beam profile on Zero Gradient Synchrotron (ZGS) in 1967 (around the same time in Budker Institute)
- Measures transverse profile of a 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.

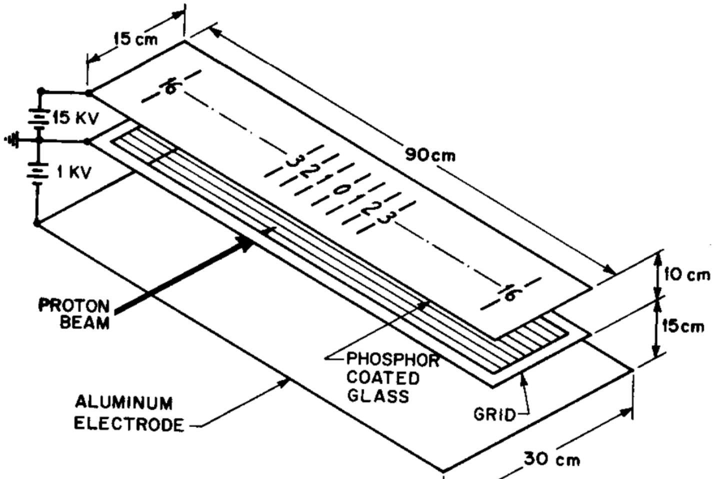


374

NONDESTRUCTIVE BEAM PROFILE DETECTION SYSTEMS
FOR THE ZERO GRADIENT SYNCHROTRON*

Fred Hornstra, Jr. and William H. DeLuca

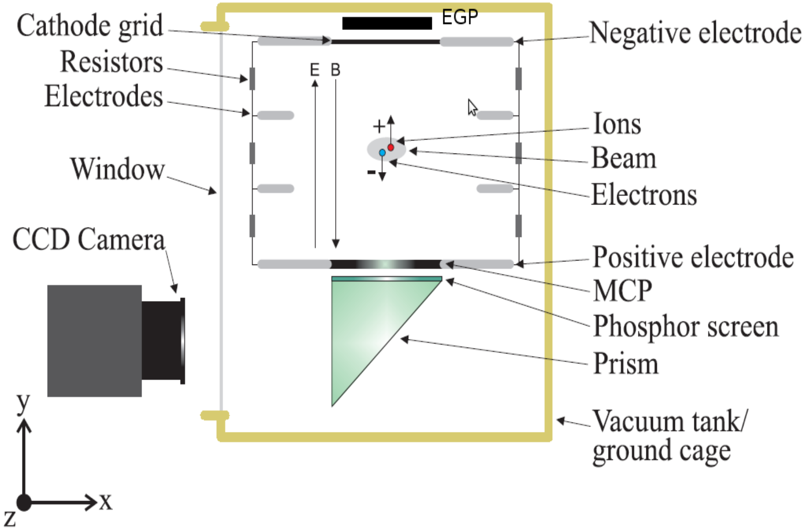
Argonne National Laboratory
Argonne, Illinois

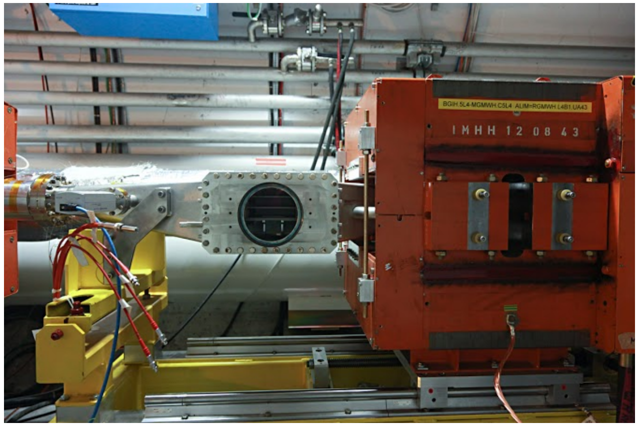


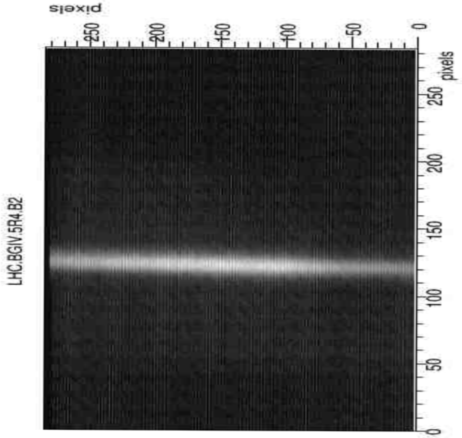
NOTE: FIDUCIAL MARKS ARE 1 INCH (2.54 cm) APART

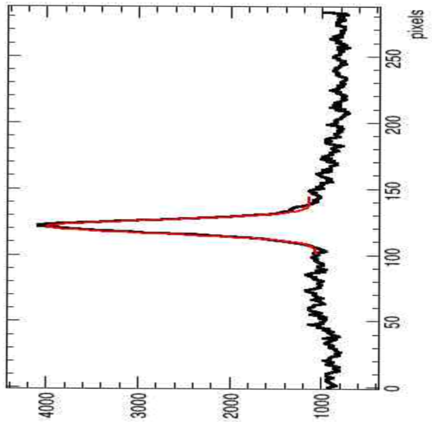
LHC/SPS IPM

- Cryogenic machine, rest gas pressure not enough: Neon injection (working pressure 10^{-8} mbar)
- Signal from ion beam much larger than proton beam
- Multi-Channel plate often saturated, gain nonuniformity after extensive use.
- Optical system PSF ≳ 150 µm





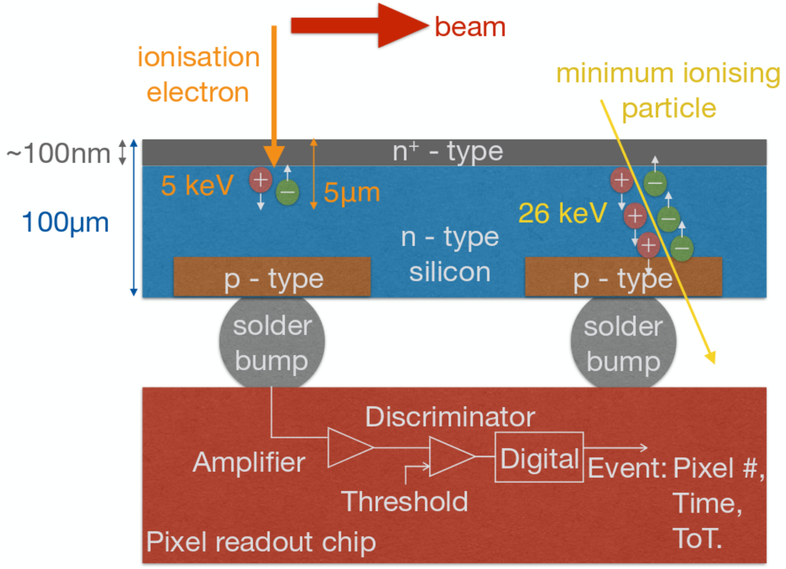




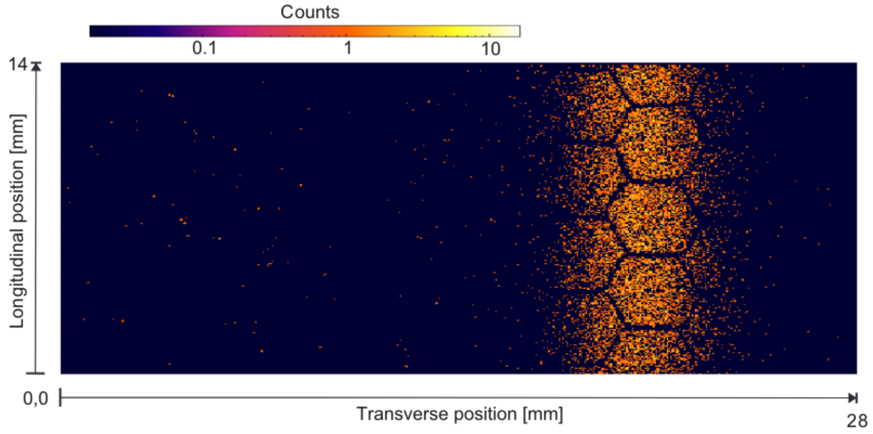
Remark:
New IPM detector technology

- Hybrid silicon pixel detector (in this case Timepix3)
- Relatively inexpensive
- Pixels 55x55 µm^2
- Single chip 256x256 pixels

- Electron arrival time resolution: 1.56 ns
- Continuous measurement
- No capricious MCP
- Prototype working well on CERN PS



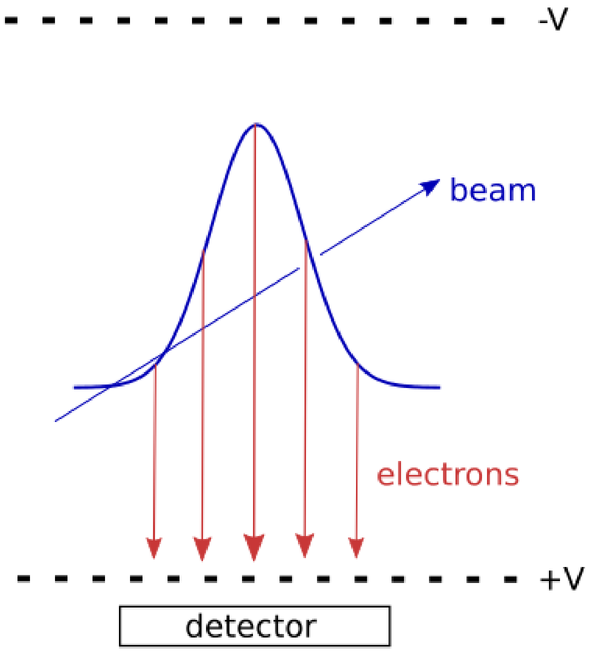
J. Storey et al., Proc. IBIC 2017(WEPCC07)
S. Levasseur et al., Proc of IPAC 2018(WEPAL075)



Profile distortion in IPM

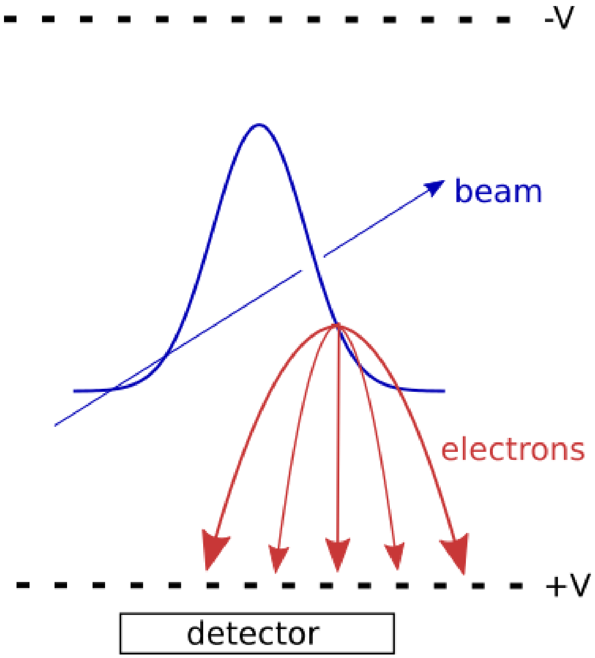
Ideal case

- 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

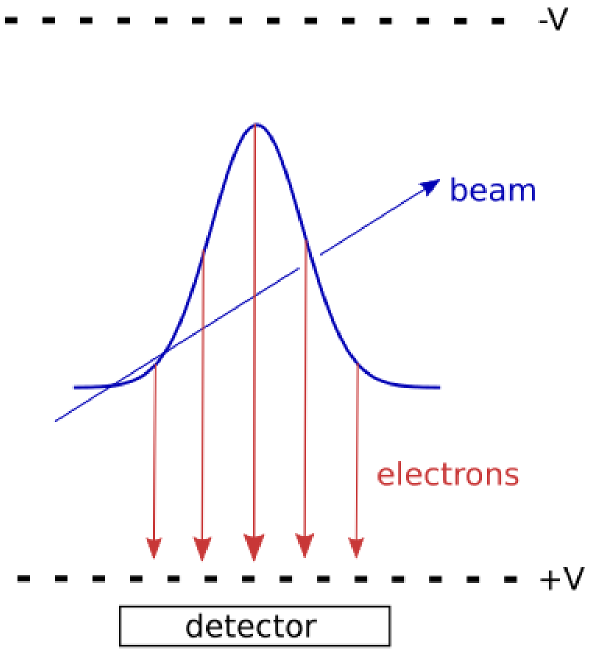


… instrumental effects such as camera tilt, optical point-spread-functions, point-spread functions due to optical system and multi-channel plate granularity etc, etc… come on top!

Profile distortion in IPM

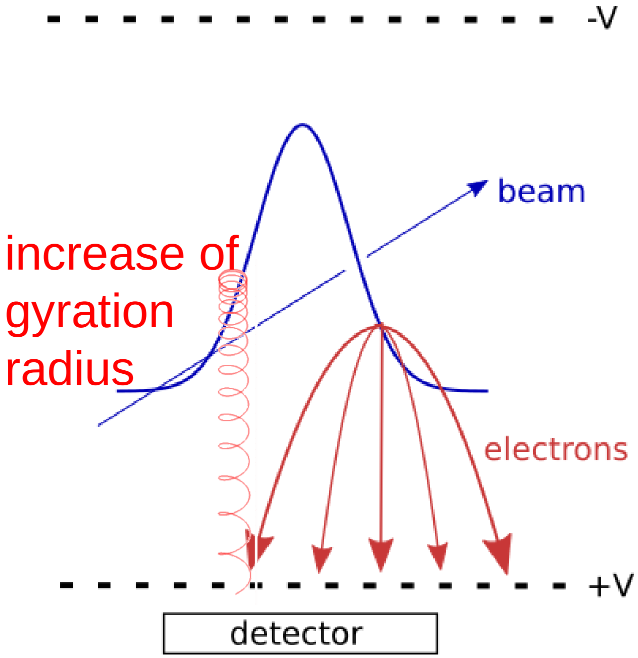
Ideal case

- 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!

Profile distortion without magnetic field (I)

- Effect already investigated in [W. DeLuca, IEEE 1969]

  (also observation of focusing when collecting electrons!)
- R.E.Thern,”Space-Charge Distortion in the Brookhaven Ionization Profile Monitor ”, PAC 1987

$$\sigma_m = \sigma + 0.302 \frac{N^{1.065}}{\sigma^{2.065}} \left(1 + 3.6\ R^{1.54}\right)^{-0.435}$$

N = beam current in 10^{12} protons

σ = root-mean-square beam size in mm.

R = aspect ratio, (other plane)/(measured plane)

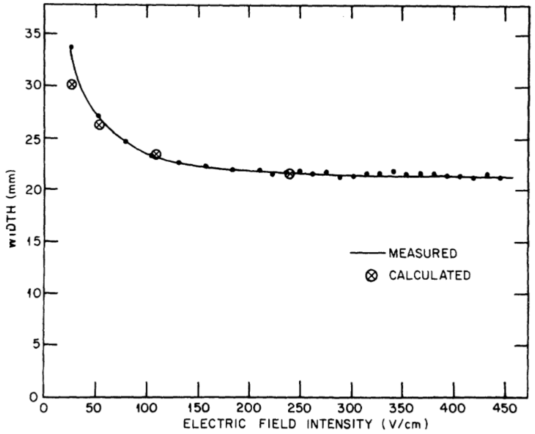


Fig. 4 Comparison of calculated and measured values of ion scatter as a function of electrode field strength.

- simulations versus measurements

- quite good agreement for nominal extraction voltages

- doubts about accuracy of the correction due to disagreement for low extraction voltages

- W. Graves, “Measurement of Transverse Emittance in the Fermilab Booster”, PhD 1994,

  - simulations with TOSCA2D

  - proposed correction:

$$\sigma_{beam} = C_1 + C_2\sigma_{measured} + C_3 N.$$

Profile distortion without magnetic field (II)

- **J. Amundson, J. Lackey, P. Spentzouris, G. Jungman, L. Spentzouris**

  ”Calibration of the Fermilab Booster ionization profile monitor”, PRSTAB 2003

  - theoretical investigations (1st order): $\langle |y_{\text{out}}| \rangle = \langle |y_{\text{in}}| \rangle + KN\sigma_{\text{real}}^{-1/2},$

  - 2D simulations using OCTAVE

  - 1st and 2nd order corrections investigated: $\sigma_{\text{measured}} = \sigma_{\text{real}} + C_1 N \sigma_{\text{real}}^{p_1} + C_2 N^2 \sigma_{\text{real}}^{p_2}.$

- **What if the beam is non-gaussian?**

  J. Egberts, “IFMIF-LIPAc Beam Diagnostics: Profiling and Loss Monitoring Systems”, PhD 2012

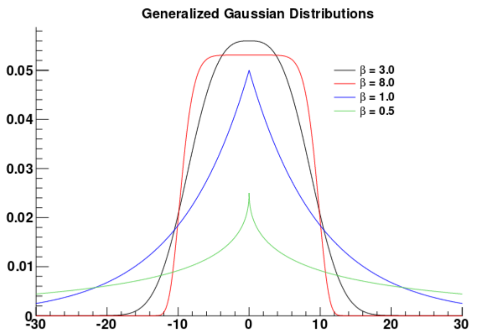
  - generalized gaussian distribution

  $$P(x) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(\frac{|x-\mu|}{\alpha})^{\beta}}$$

  - correction via matrix multiplication:

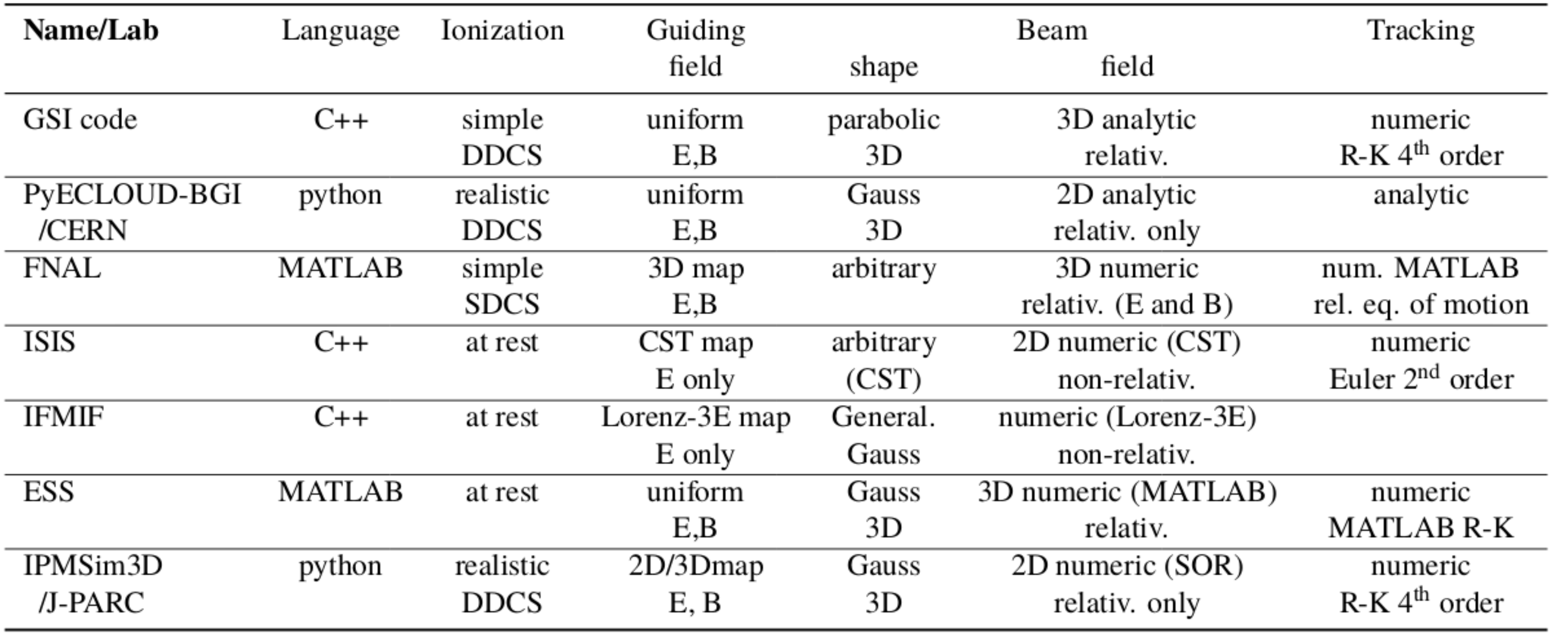
  $P_{\text{corrected, I}} = \Sigma_j \, (A_{\,i,j} \, P_{\text{measured},j})$, where A=A(N, σ_{measured}, κ-kurtosis)

  - if solution not found immediately – iterative, convergent process.



IPM simulation programs

- Since ~2012 – looking for proper IPM simulation codes
- Available programs (CST, Geant4) – missing features
- Many ‘private’ codes exists.
- Workshops on IPM simulations: CERN 2016, GSI 2017, J-PARC 2018 (Sep)

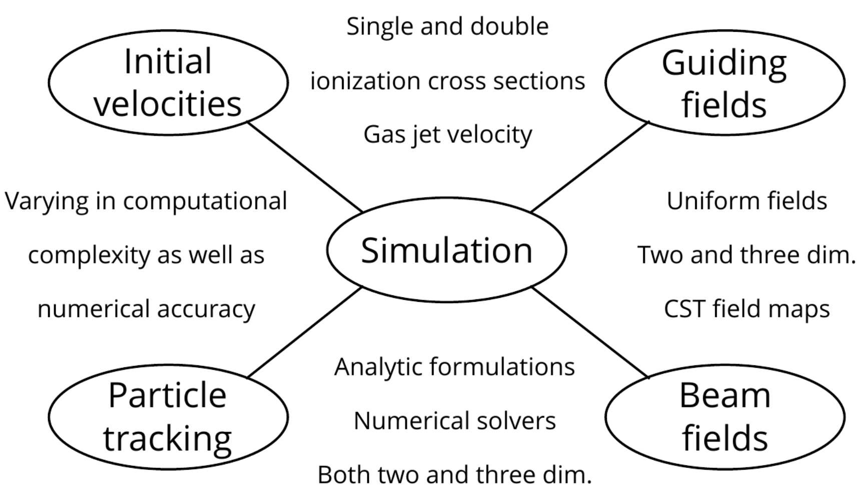


Name/Lab	Language	Ionization	Guiding		Beam	Tracking
			field	shape	field	
GSI code	C++	simple DDCS	uniform E,B	parabolic 3D	3D analytic relativ.	numeric R-K 4th order
PyECLOUD-BGI /CERN	python	realistic DDCS	uniform E,B	Gauss 3D	2D analytic relativ. only	analytic
FNAL	MATLAB	simple SDCS	3D map E,B	arbitrary	3D numeric relativ. (E and B)	num. MATLAB rel. eq. of motion
ISIS	C++	at rest	CST map E only	arbitrary (CST)	2D numeric (CST) non-relativ.	numeric Euler 2nd order
IFMIF	C++	at rest	Lorenz-3E map E only	General. Gauss	numeric (Lorenz-3E) non-relativ.	
ESS	MATLAB	at rest	uniform E,B	Gauss 3D	3D numeric (MATLAB) relativ.	numeric MATLAB R-K
IPMSim3D /J-PARC	python	realistic DDCS	2D/3Dmap E, B	Gauss 3D	2D numeric (SOR) relativ. only	numeric R-K 4th order

M. Sapinski et al, *Ionization Profile Monitor Simulations - Status and Future Plans*, Proc. of IBIC 2016, (TUPG71)

Virtual-IPM program

- After looking for a proper program we decided to write our own called Virtual-IPM
- Written in Python with modern, modular architecture
- GUI in Qt
- Covers: IPM, BIF, gas jets
- Publicly available as python module: https://pypi.org/project/virtual-ipm
- pip install virtual-ipm
- Code on gitlab: https://gitlab.com/IPMsim/Virtual-IPM



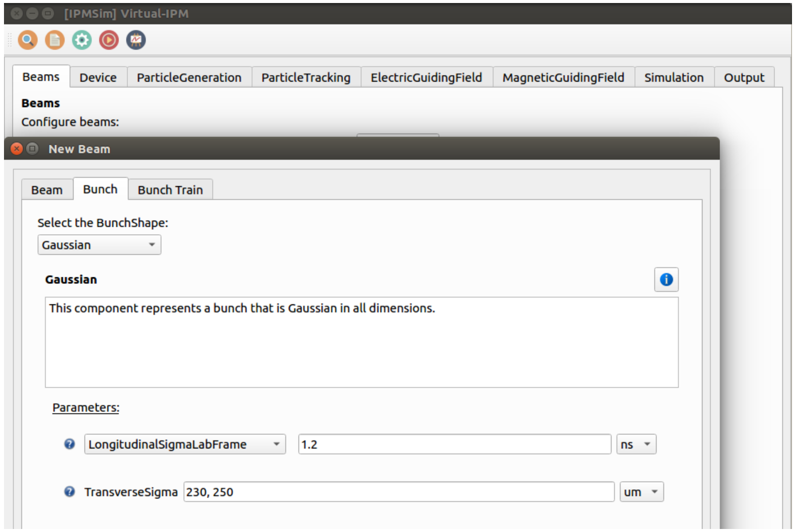
Simulation

Initial velocities: Single and double ionization cross sections; Gas jet velocity

Guiding fields: Uniform fields; Two and three dim.; CST field maps

Particle tracking: Varying in computational complexity as well as numerical accuracy

Beam fields: Analytic formulations; Numerical solvers; Both two and three dim.



[IPMSim] Virtual-IPM

Beams | Device | ParticleGeneration | ParticleTracking | ElectricGuidingField | MagneticGuidingField | Simulation | Output

Beams

Configure beams:

New Beam

Beam | Bunch | Bunch Train

Select the BunchShape: Gaussian

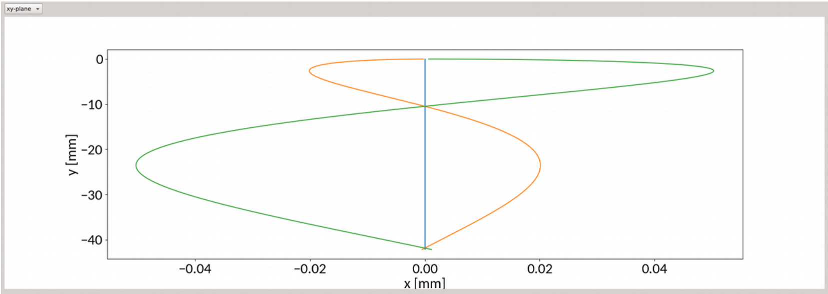
Gaussian

This component represents a bunch that is Gaussian in all dimensions.

Parameters:

LongitudinalSigmaLabFrame: 1.2 ns

TransverseSigma: 230, 250 um



D. Vilsmeier, et al., *A Modular Application for IPM Simulations*, Proc. of IBIC17 (WEPCC07)
and presentation at 1st ARIES Annual Meeting, Riga, May 2018

High space-charge regime: LHC beams

- Let’s fix beam parameters the impact of various phenomena

  - for this beam maximum E-field is ~MV/m

  - we do simulate beam B-field, but its impact is small (~10 mT and electrons/ions move slowly)

- IPM corresponding to LHC/SPS IPMs but with Timepix3 detector resolution:

  - distance between electrodes: 84 mm

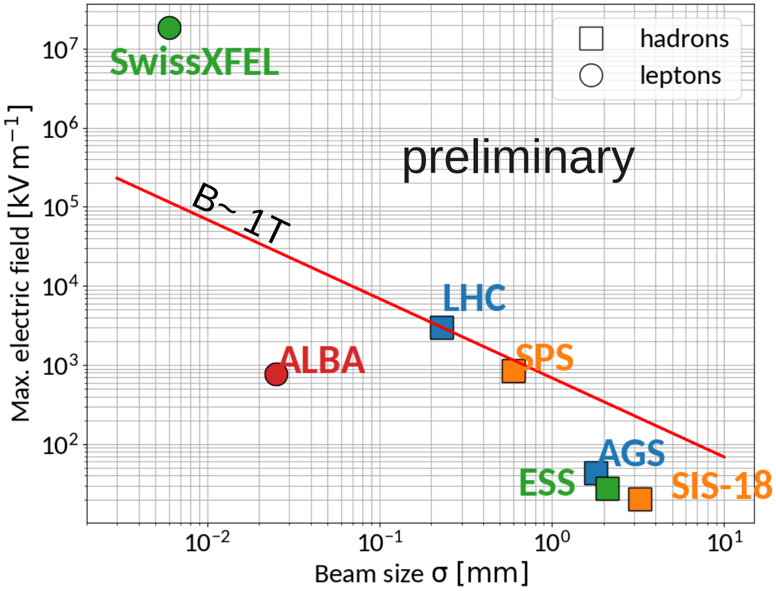
  - U=4 kV (E=48 kV/m)

  - B=0.2 T

  - Spatial resolution (binning): 55 µm

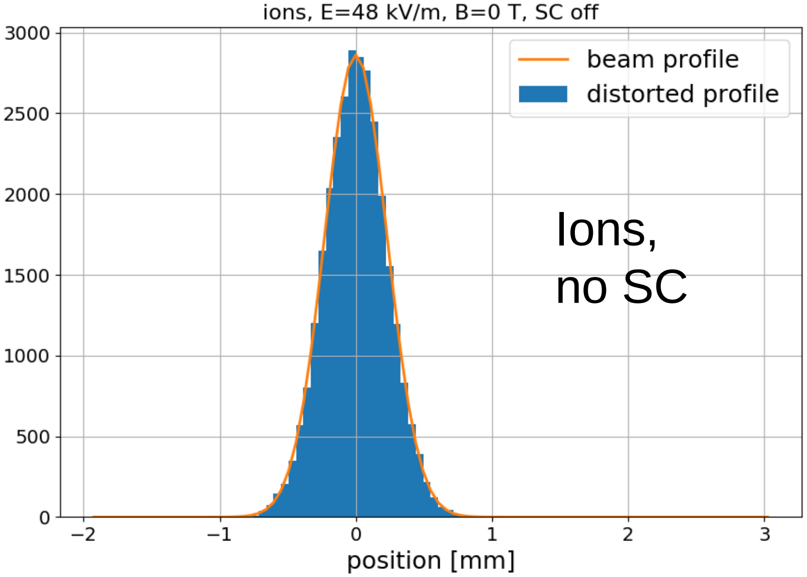
Here “space charge” refers to bunch field impact on electrons – it scales with relativistic gamma.

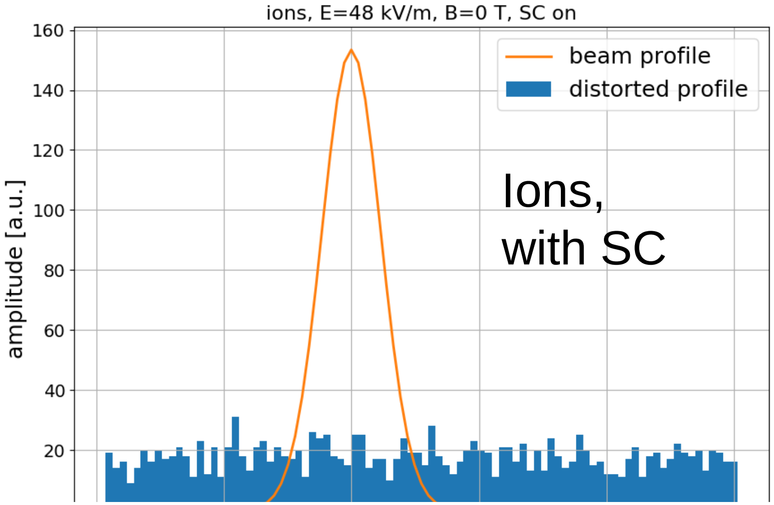
σ_x	230 µm
σ_y	270 µm
N_{prot}	$1.4 \bullet 10^{11}$
$4\sigma_z$	1.1 ns
E_{beam}	6.5 TeV

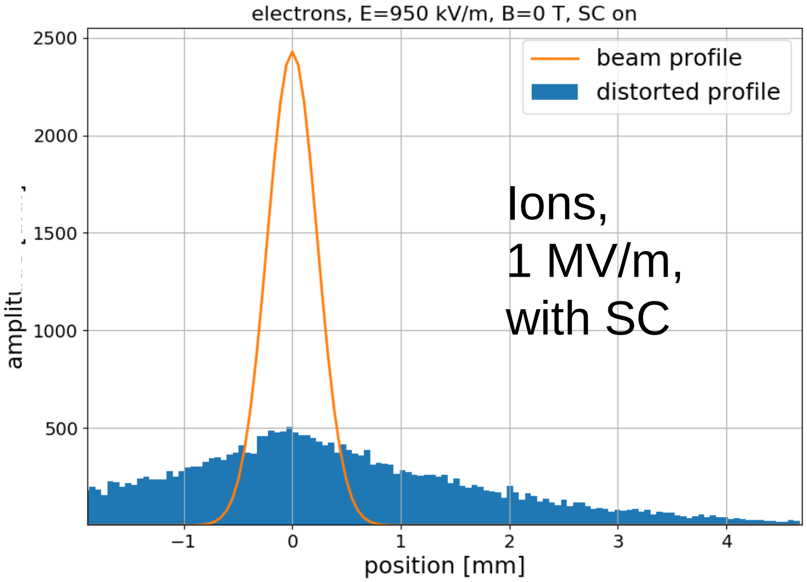


Understanding distortion: ions

- The simplest device (no magnetic field)
- Tracked: H_2^+ ions



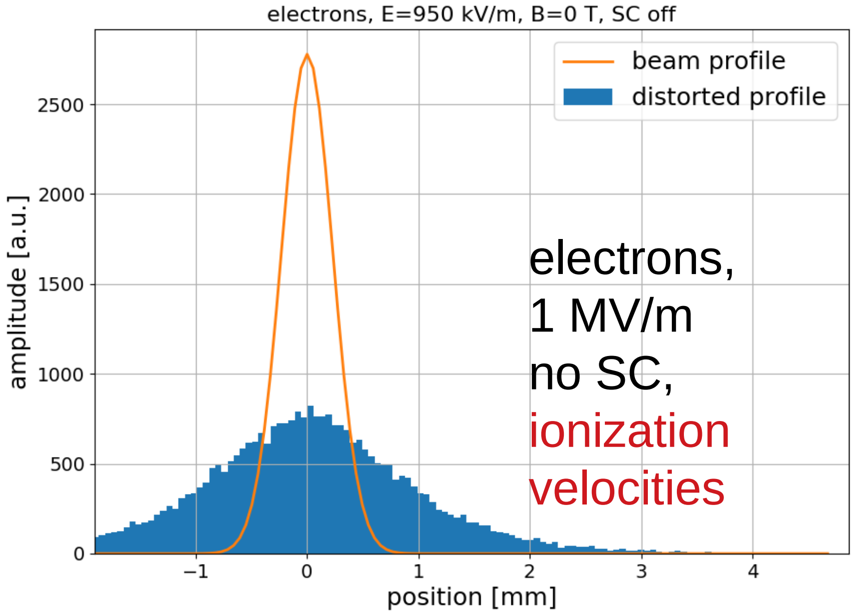




Conclusion: ion-based device too sensitive to beam space charge → use electrons!

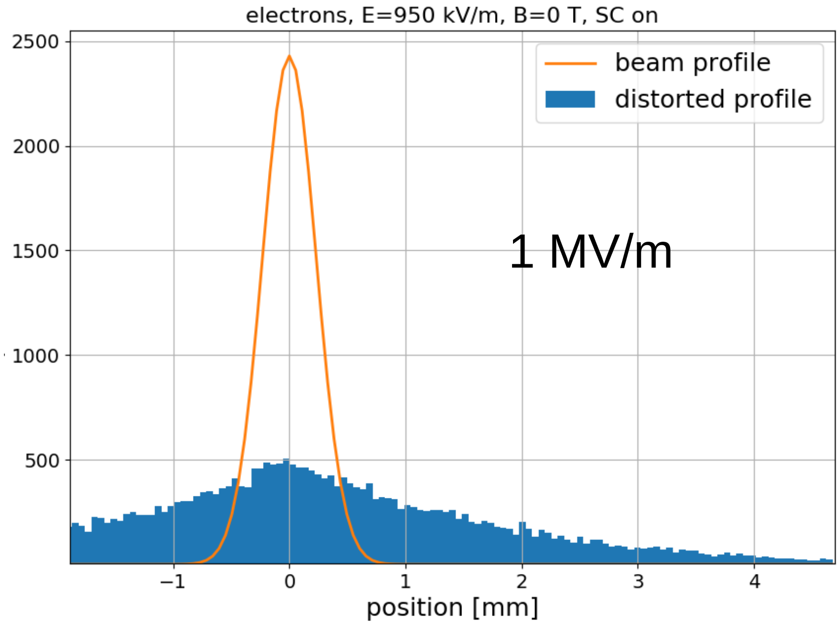
Electron tracing – no B-field

- Immediately try higher extraction fields (~ 1 MV/m)



electrons, E=950 kV/m, B=0 T, SC off

electrons,
1 MV/m
no SC,
ionization
velocities



electrons, E=950 kV/m, B=0 T, SC on

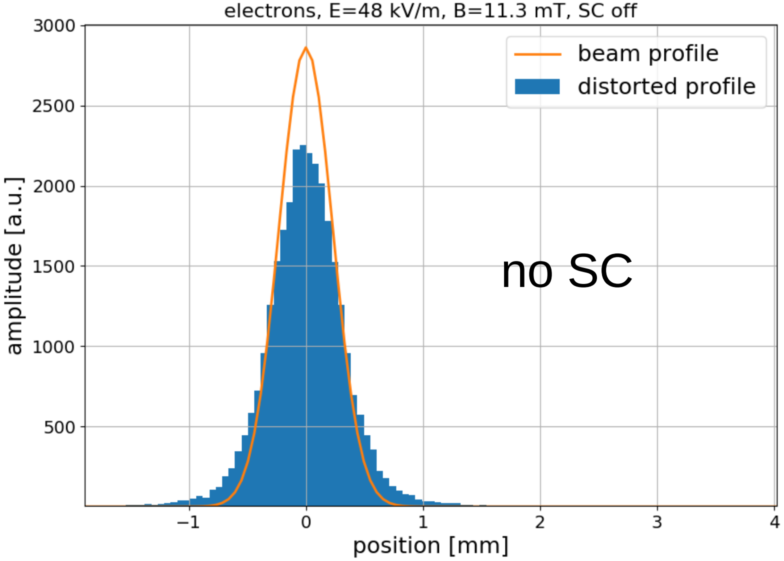
1 MV/m

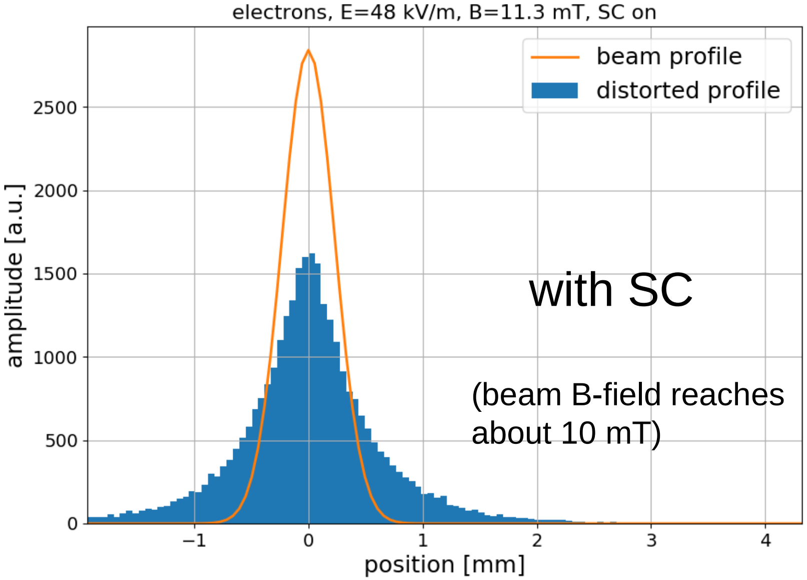
Space charge smears the profile, but even without space charge, the initial electron velocities from ionization broaden the peak enormously.

Magnetic field is needed when using electrons with sub-mm beams.

Electron tracing – *single turn* B-field

- Magnetic field tuned to single turn: $B=\pi\sqrt{2 * m_e * E/(e * d)}$= 11.3 mT
- Original idea in [F. Hornsta, M. Trump, PLACC 1970]



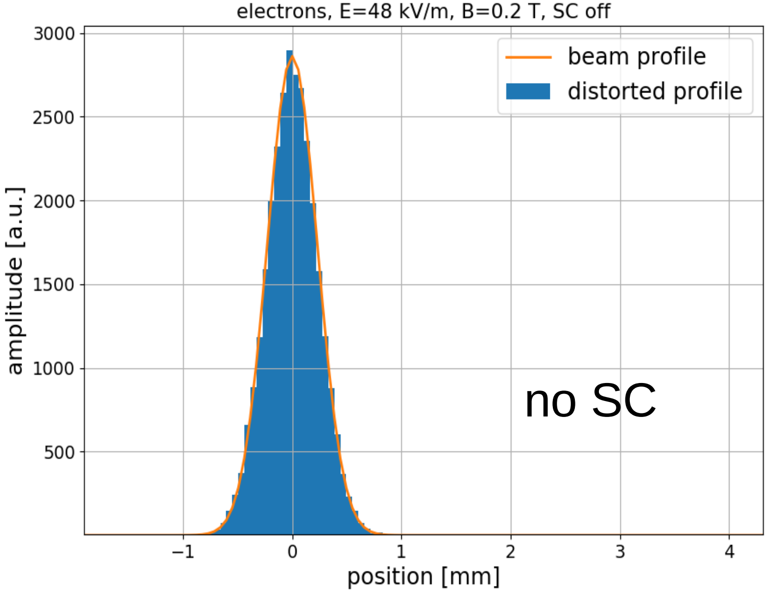


(beam B-field reaches about 10 mT)

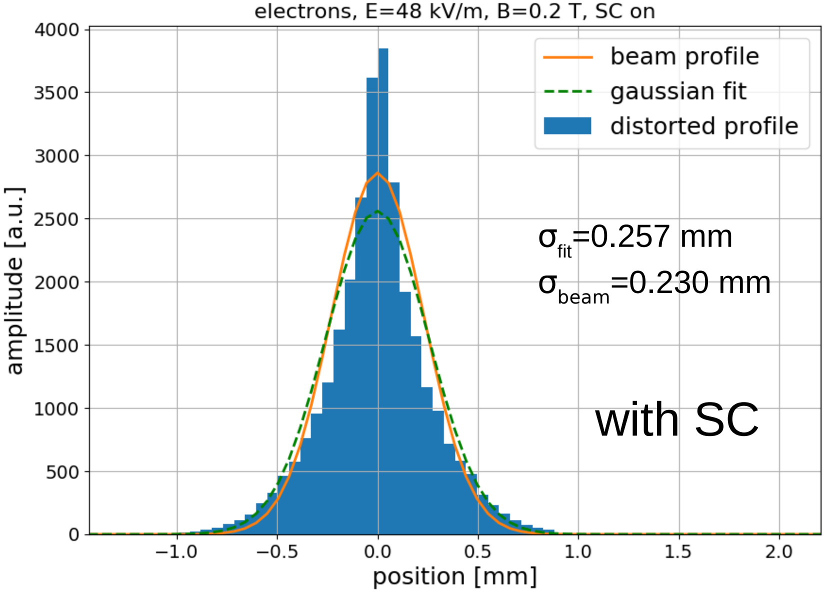
Disadvantages of tuning magnetic field to single turn:

- Sensitivity to vertical beam position
- Sensitivity to fringe fields of neighboring magnets (as the field required is weak)
- Spread due to vertical component of electron velocities (and vertical beam size)
- Little help for the space-charge effects – electron trapping/additional kick in vertical direction

Electron tracing – strong B-field


electrons, E=48 kV/m, B=0.2 T, SC off

no SC


electrons, E=48 kV/m, B=0.2 T, SC on

σ_{fit}=0.257 mm

σ_{beam}=0.230 mm

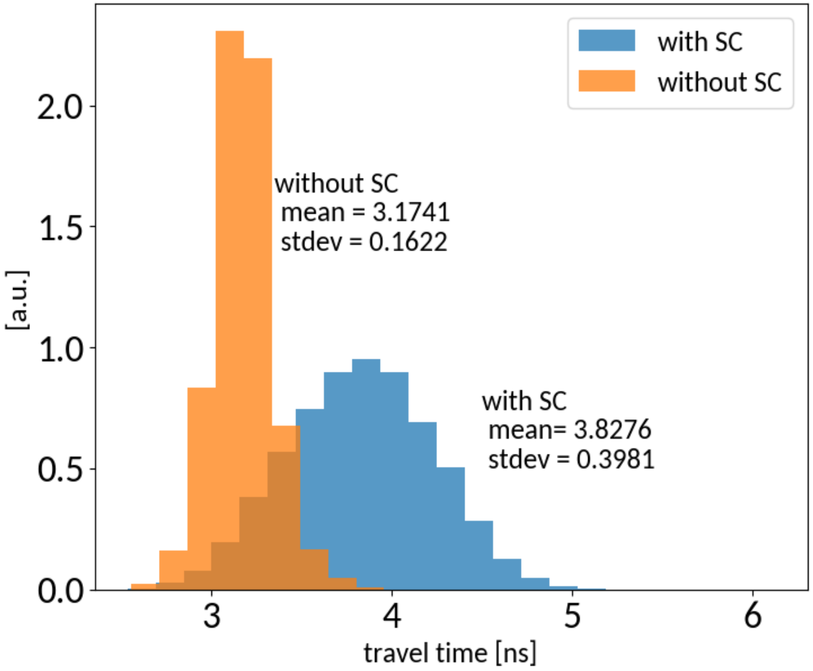
with SC

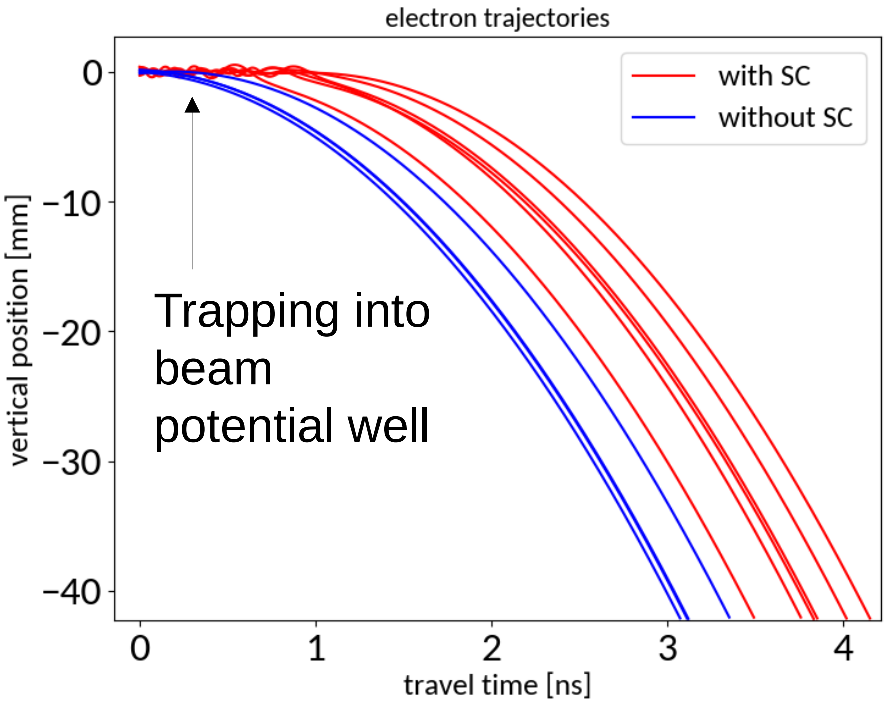
Conclusions:

- Better use strong B-field rather than “one turn” field
- 0.2 T is not enough to counteract space charge effect for LHC beam.

Electron tracing – strong B-field

What happens to electrons?



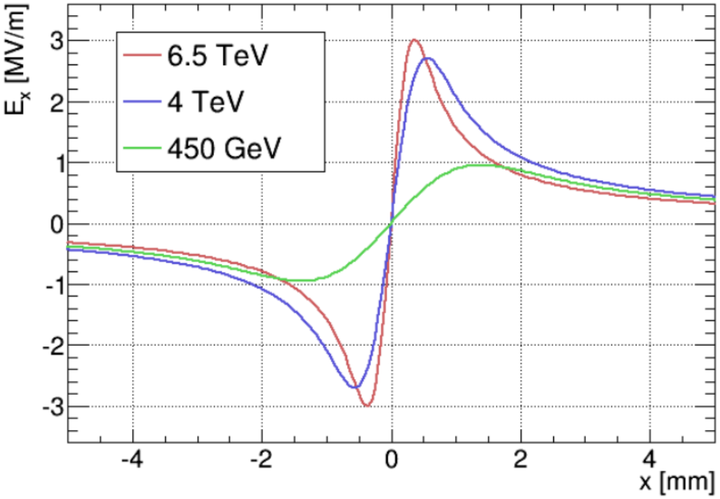


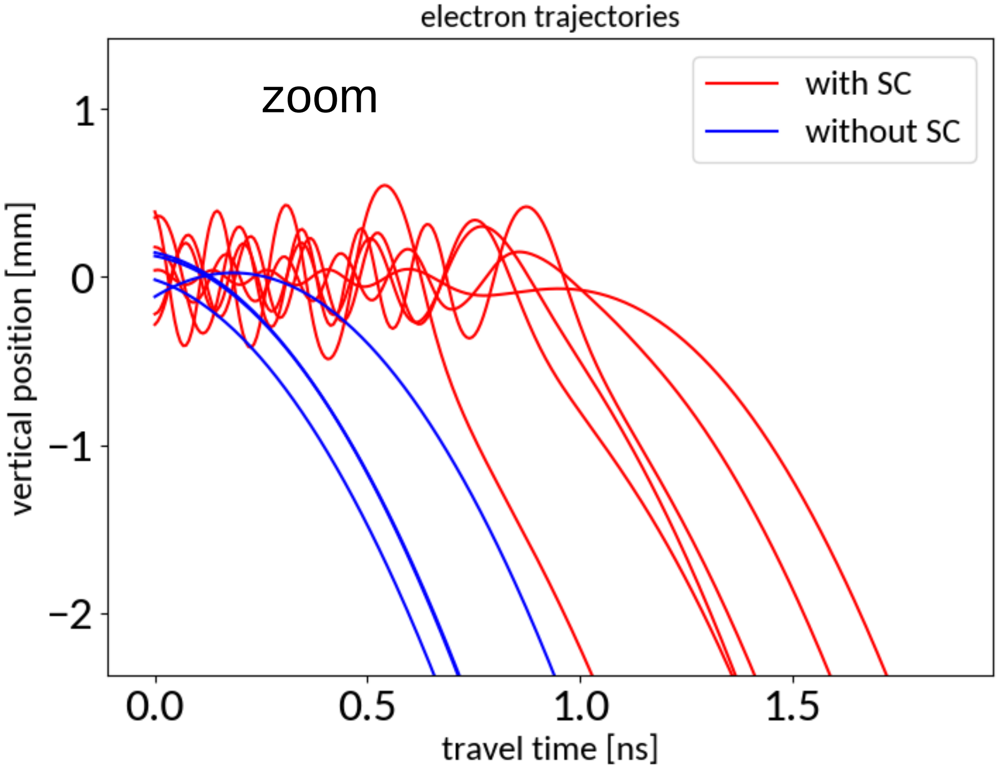
Trapping into beam potential well

Electrons are trapped in bunch field for the time when bunch passes. They make several oscillations around bunch center. Complex movement!

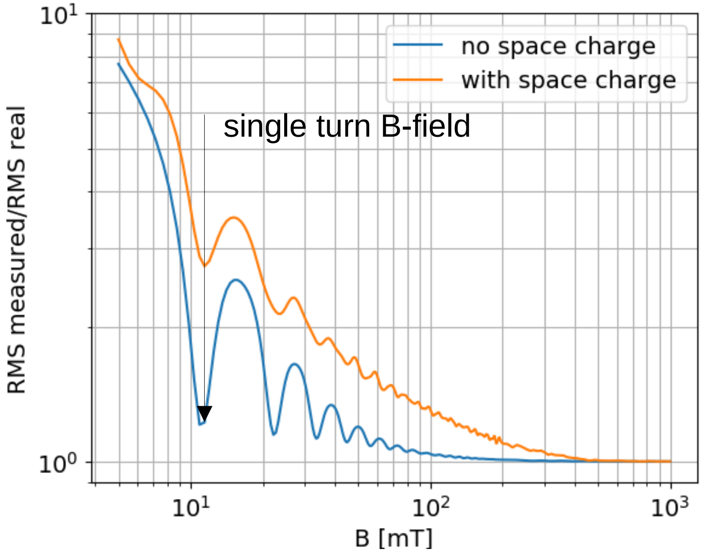
Electron tracing – strong B-field

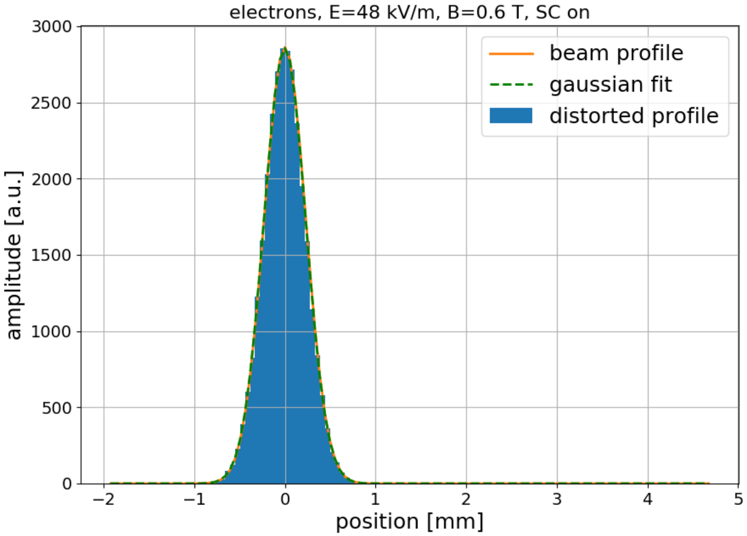
More investigations – what happens to electrons?





Electron tracing – very strong B-field



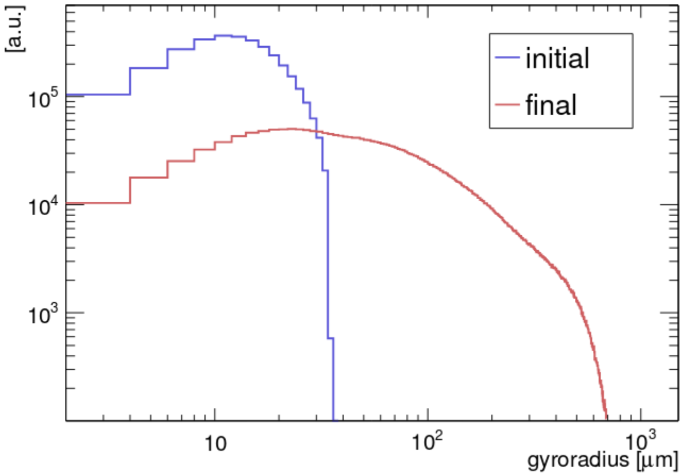


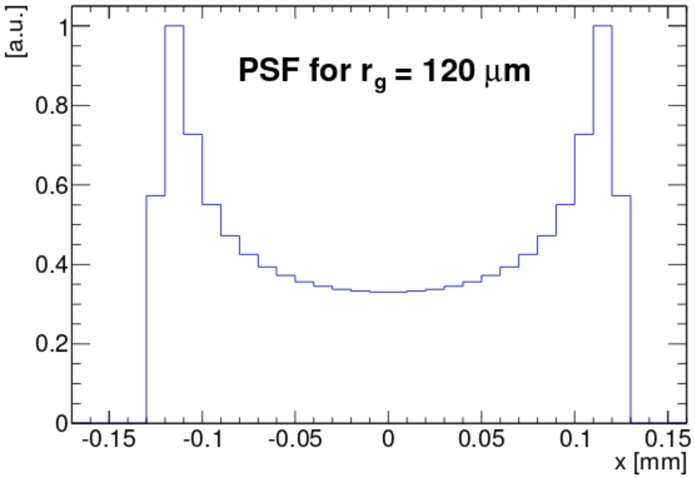
- unpractical solution: big and expensive magnet!

Correction methods: sampling gyroradius

- No simple analytic procedure/formula (as for electric field case) found
- **Electron sieve**
  - MS. et al., *Investigation of the effect of beam space-charge on electrons in ionization profile monitors*, Proc. of HB2014 (MOPAB42)
  - D. Vilsmeier, CERN-THESIS-2015-035

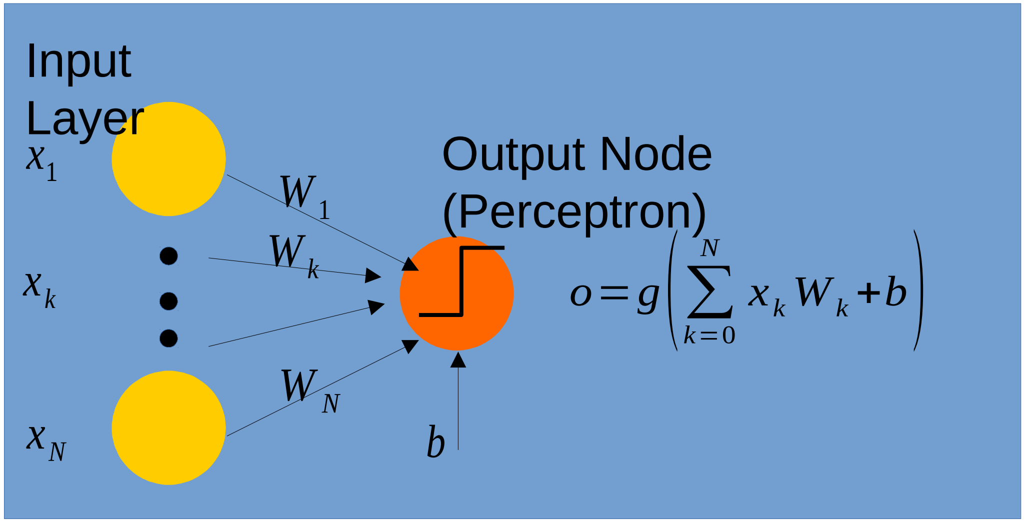
  - technically difficult
- Methods based on Machine Learning algorithms: finding arbitrary mapping between distorted profile and original one.
  - R. Singh, M. S., D. Vilsmeier, *Simulation supported profile reconstruction with machine learning*, Proc. of IBIC17 (WEPCC06)
  - D. Vilsmeier et al., *Reconstructing Space-Charge Distorted IPM Profiles with Machine Learning Algorithms*, Proc. of IPAC 2018 (WEPAK008)





Artificial Neural Network

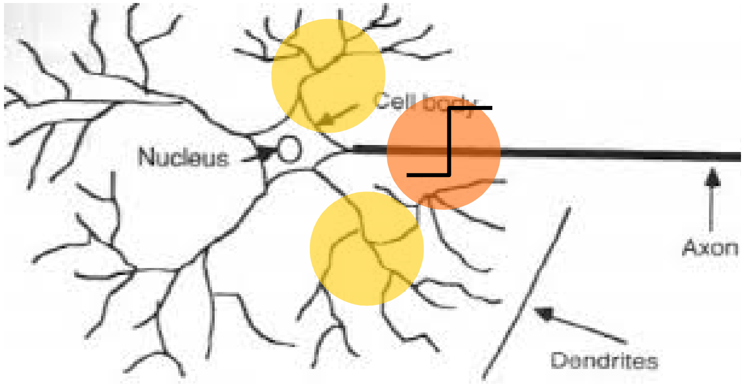
- Machine Learning - algorithms which can learn and make predictions on data, without explicit programming
- Biologically inspired → Brain cells -> neurons, computation via connections and thus Networks
- The basic node of ANNs is “Perceptron”

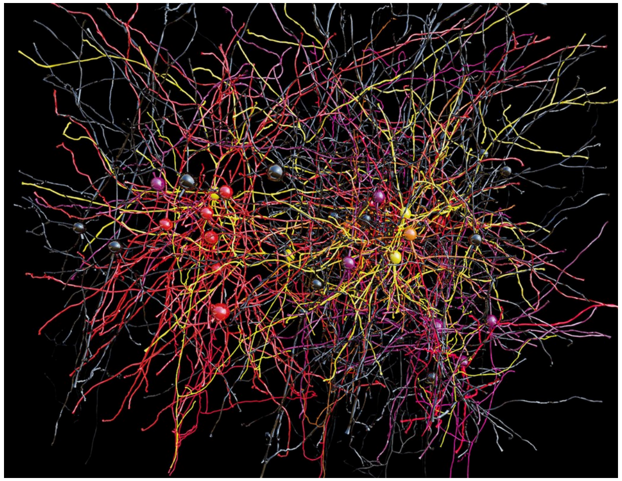


Input Layer

Output Node (Perceptron)

$$o = g\left(\sum_{k=0}^{N} x_k W_k + b\right)$$



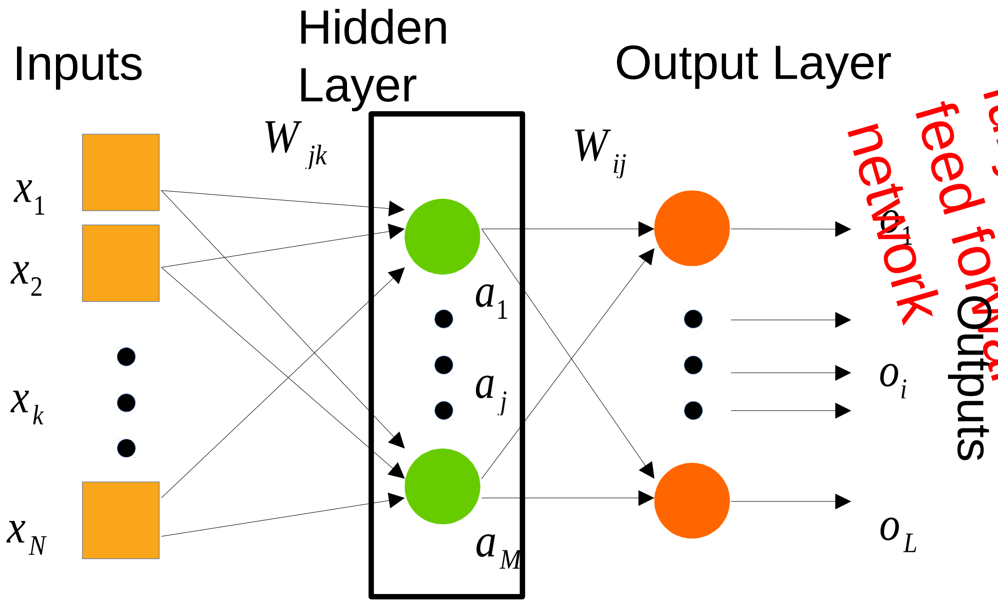


[https://www.wired.com/2016/03/took-neuroscientists-ten-years-map-tiny-slice-brain]

Perceptron parameters:

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

Hidden layers



Multi-layer Perceptron or fully connected 2-layer or feed forward neural network

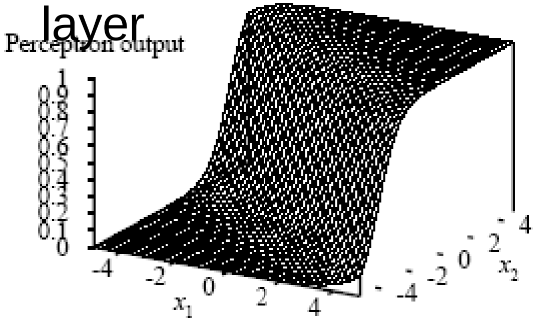
- Each hidden layer and output layer node is a perceptron

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

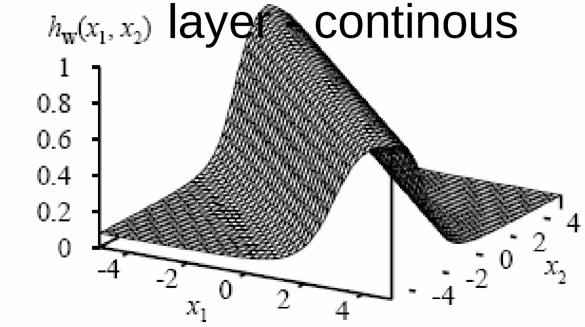
Adding “hidden” layer(s) allow non-linear target functions to be represented

Multi-layer perceptron (MLP)

Perceptron: No hidden layer



One hidden layer - continous



(In practice we use additional hidden layers even to continous problems)

- 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, $y=f(x,y,z\ldots)$ it can be approximated by:

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

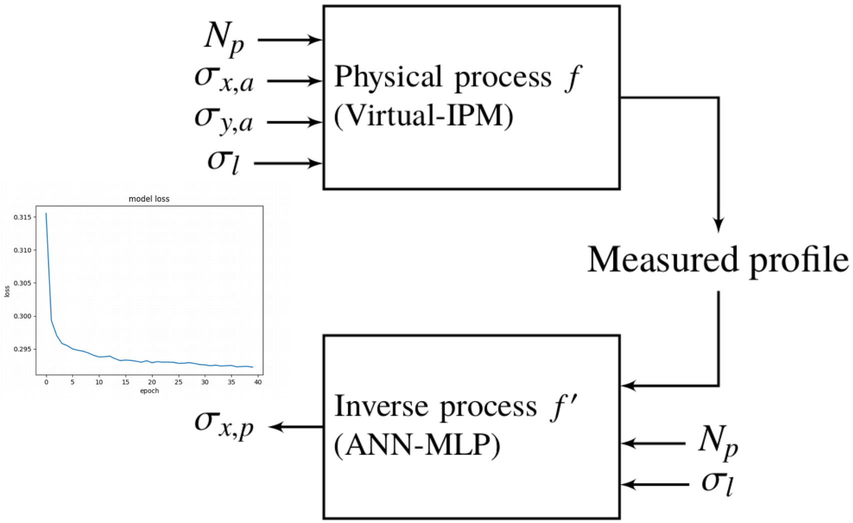
universal approximator

Profile correction using ANN (I)

First approach:

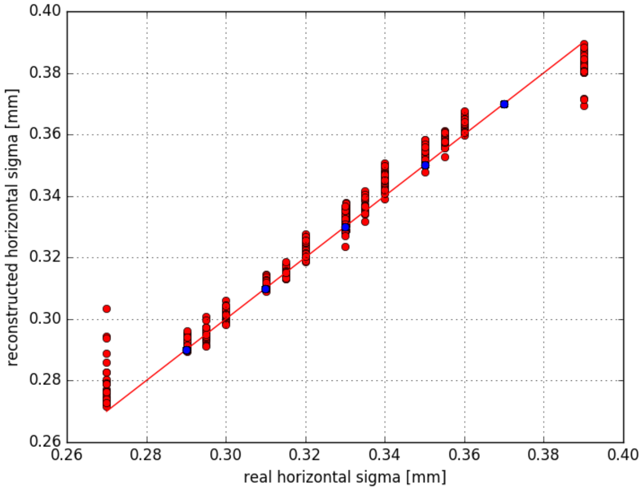
- Training sample on grid (375 points):

σ_x[mm]	0.29, 0.31, 0.33, 0.35, 0.37
σ_y[mm]	0.4,0.45,0.5,0.55,0.6
N_p [10^{11}]	1.1, 1.25, 1.40, 1.55, 1.7
4*σ_z[ns]	0.9, 1.05, 1.2



- 2-layer network, 4 validation samples:
  - 1%, 25%, 50% off
  - 100% off – outside grid
- Convolute with Point Spread Function
- Use *tensorflow* and *Matlab NN toolbox*
- Value of σ_x restored with 1% accuracy!
- Good performance with noise.
- R. Singh, M. Sapinski, D. Vilsmeier, *Simulation supported profile reconstruction with machine learning*,

  Proc. of IBIC17 (WEPCC06)

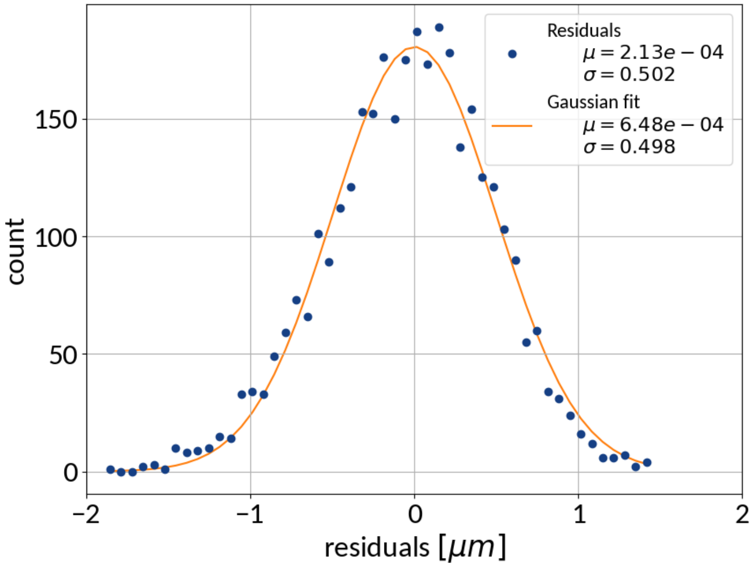


Profile correction using ANN (II)

Second approach:

- Training: sample of 13500 random points
- 4-layer network
- Validation set – also random
- Verify also other Machine Learning algorithms:

- linear regression,
- kernel Ridge regression,
- support vector machine



D. Vilsmeier et al., *Reconstructing Space-Charge Distorted IPM Profiles with Machine Learning Algorithms*, Proc. of IPAC 2018, paper WEPAK008

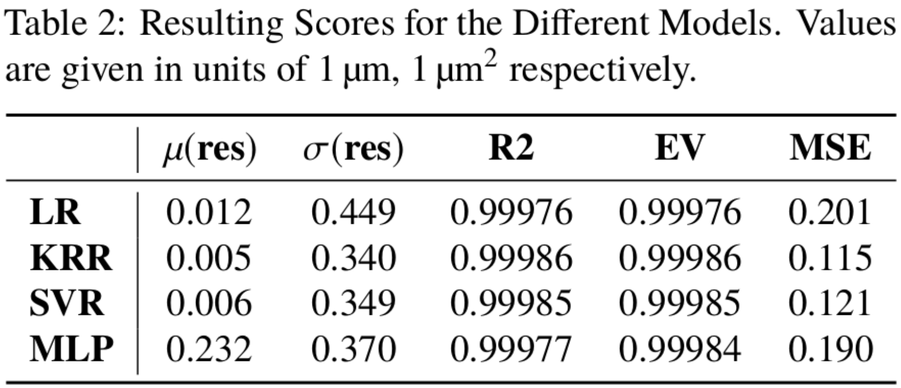


Table 2: Resulting Scores for the Different Models. Values are given in units of 1 µm, 1 µm^2 respectively.

	μ(res)	σ(res)	R2	EV	MSE
LR	0.012	0.449	0.99976	0.99976	0.201
KRR	0.005	0.340	0.99986	0.99986	0.115
SVR	0.006	0.349	0.99985	0.99985	0.121
MLP	0.232	0.370	0.99977	0.99984	0.190

Surprisingly: even linear regression gives very good results.

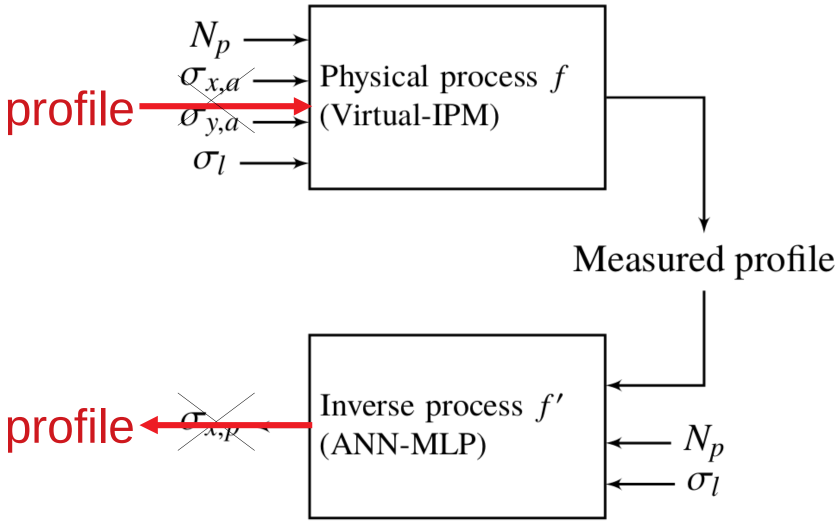
But we have not studied noise here

Profile reconstruction using ANN

Third approach:

- Training: sample of 13500 random points (only gaussian profiles)
- 2-layer network
- Validation set – also random, gaussian and non-gaussian
- Profile: 98 bins, 55 μm/bin
- Quality of profile reconstruction assessed using Mean Squared Deviations between real beam profile and corrected profile

- Results for gaussian profiles:
  Very good profile shape reconstruction



N_p

$\cancel{\sigma_{x,a}}$

profile

$\cancel{\sigma_{y,a}}$

σ_l

Physical process f
(Virtual-IPM)

Measured profile

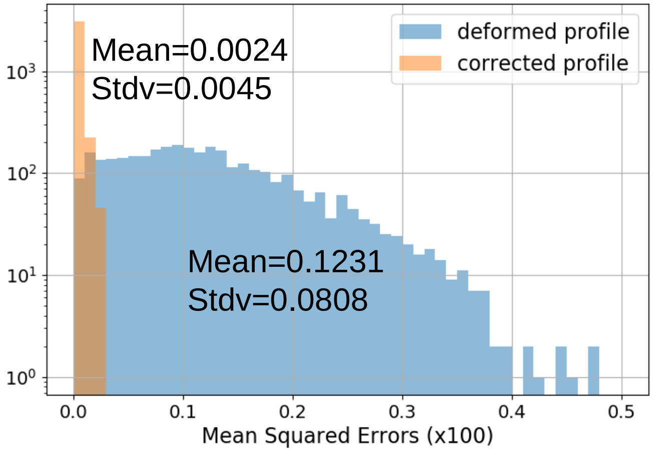
Inverse process f'
(ANN-MLP)

N_p

σ_l

profile

$\cancel{\sigma_{x,p}}$



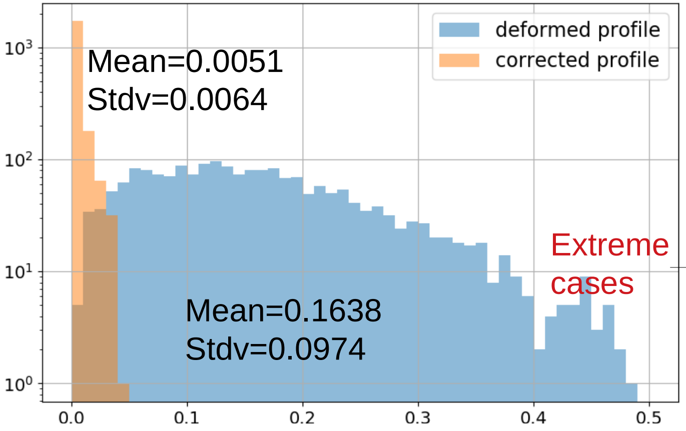
Mean=0.0024
Stdv=0.0045

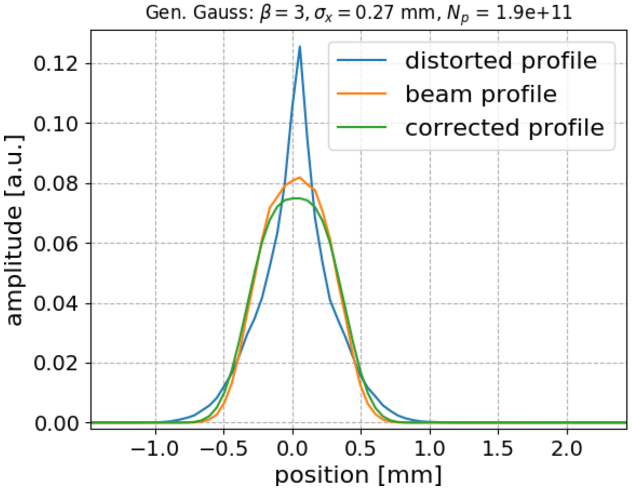
Mean=0.1231
Stdv=0.0808

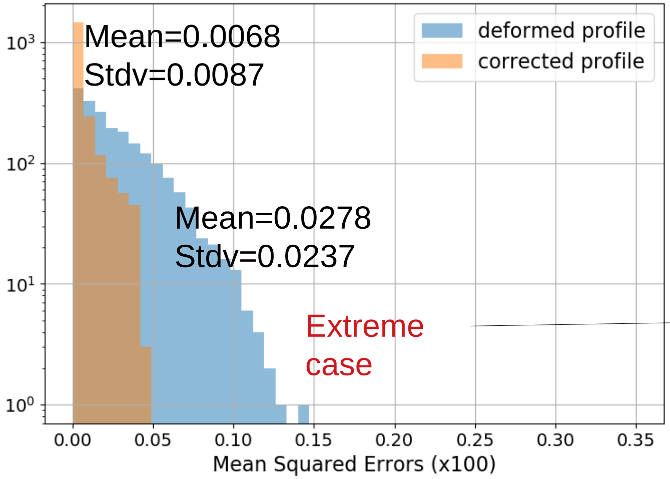
Reconstructing the profile: generalized gaussian

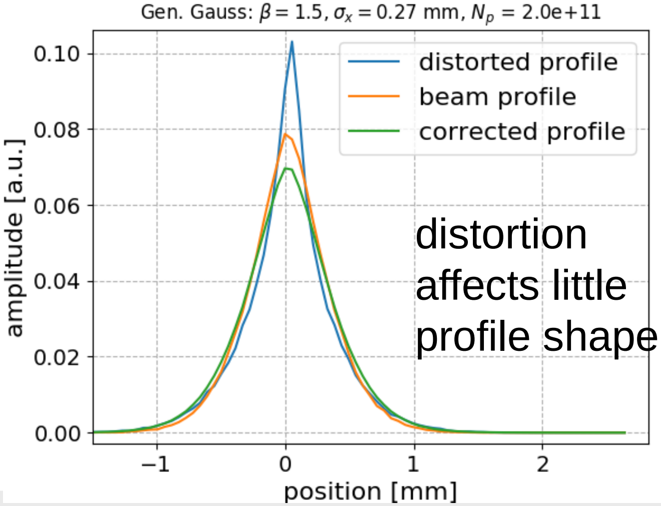
$$\frac{\beta}{2\alpha\Gamma(1/\beta)}\, e^{-(|x-\mu|/\alpha)^{\beta}}$$

- Generalized-gaussian profiles with beta=3 and 1.5
- Neural network trained only on gaussian profiles!







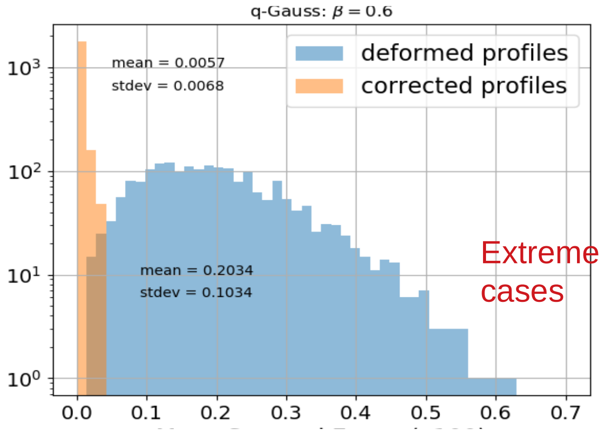


distortion affects little profile shape

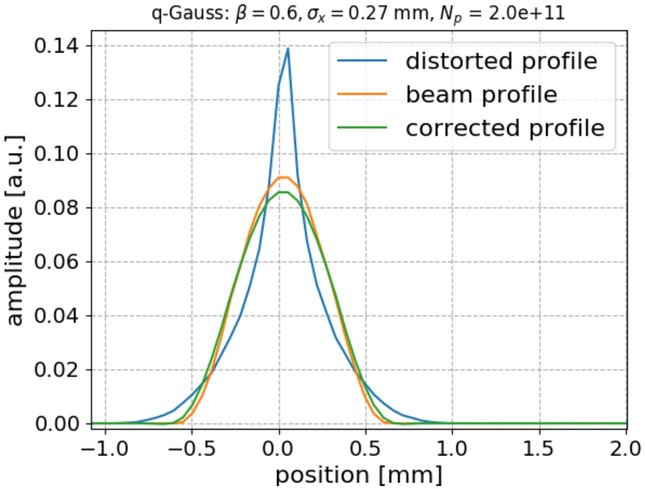
Reconstructing the profile: q-Gaussian

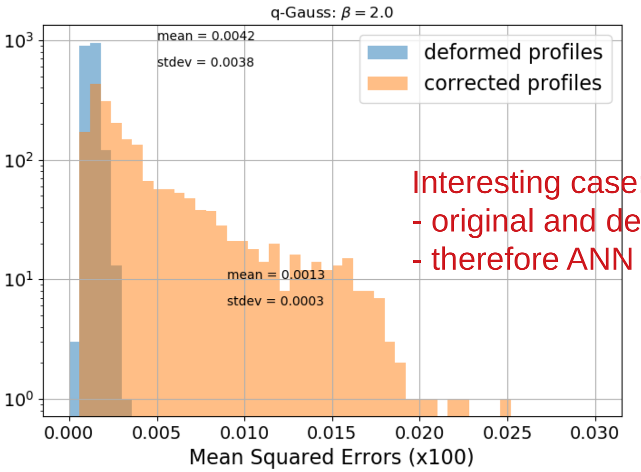
$$\frac{\sqrt{\beta}}{C_q} e_q(-\beta x^2)$$

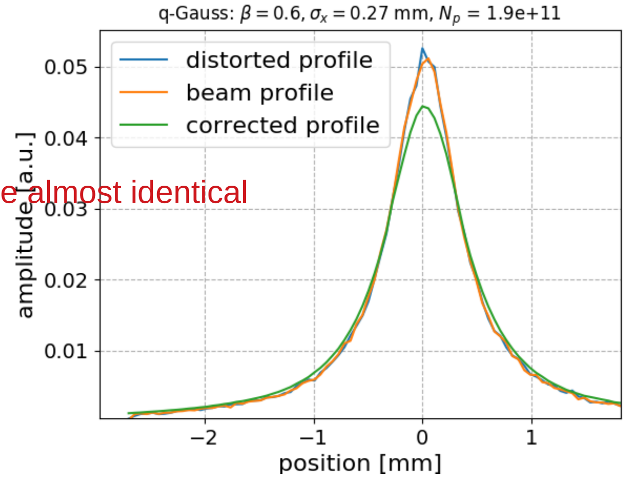
- Used for beam halo parametrization, beta=0.6 and 2.0
- Neural network trained only on gaussian profiles!



Extreme cases



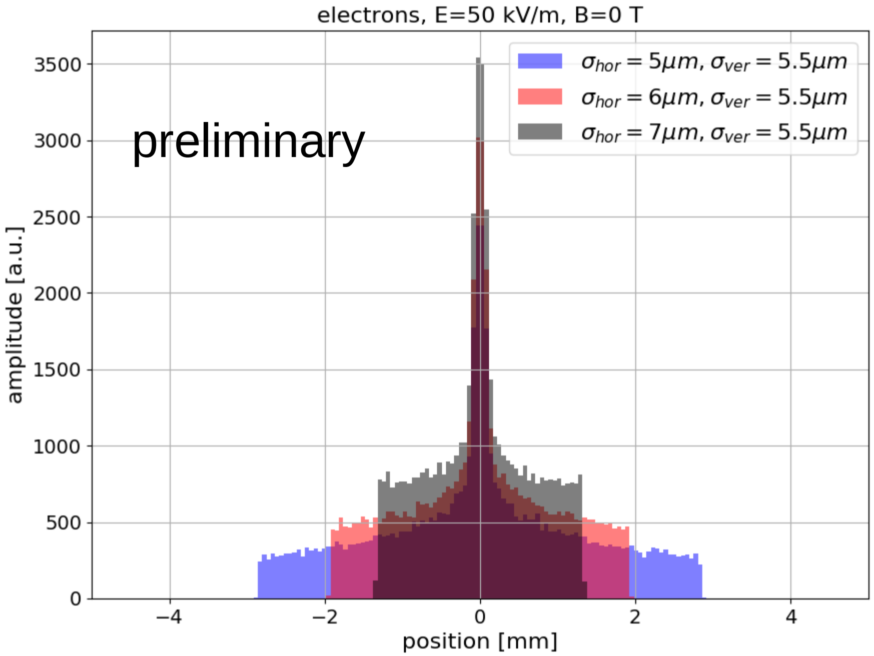




Interesting case:
- original and deformed profile are almost identical
- therefore ANN does not work!

Remark:
measuring micrometer-size beams

- If we understand the beam profile deformation, we could use it to measure high-brightness beams smaller than the resolution of the detector.
- Example SwissXFEL: 5.8 GeV electron beam, 230 pC bunch charge, 21 fs bunch length, 5-7 µm transverse size.
- Even if bunch size is 1/10th of detector resolution, the shape of the deformed profile strongly depends on the bunch size!
- Alternative to *R. Tarkeshian et al. Phys. Rev. X 8, 021039* (beam width reconstruction based on ion energies)

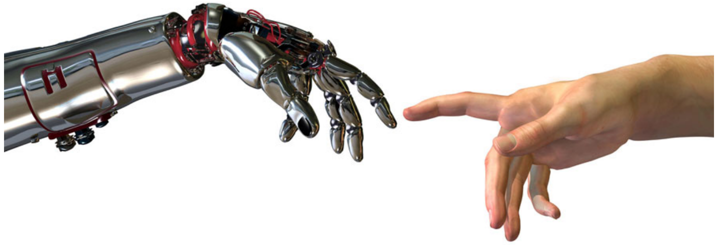


Conclusions

- Beam size measurement of bright beams is challenging
- Even magnetic IPMs suffer from measured profile deformation
- Machine Learning algorithms perform very efficient profile correction

  (network learns about nature of space-charge deformation, not just about transformation of gaussian profile)
- Sometimes using ANN an overkill, try simpler techniques (linear regression)
- Modern tools (eg. tensorflow+keras, sklearn) are easy to use
- Some physicists are skeptical because “black box” nature of ML and

  lack of error estimation



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
- …

Acknowledgments:

A. Reiter, P. Forck, K. Sato

Additional slides

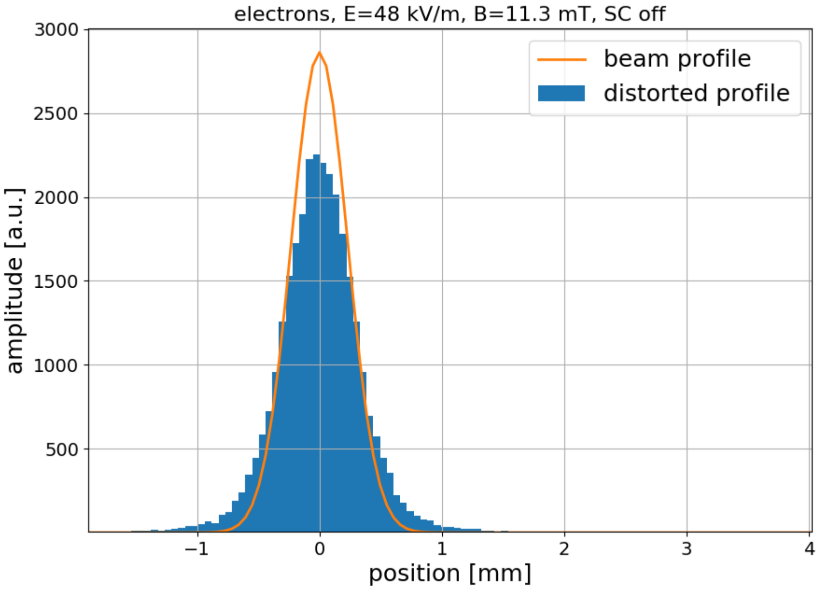
MLP Network training (I)

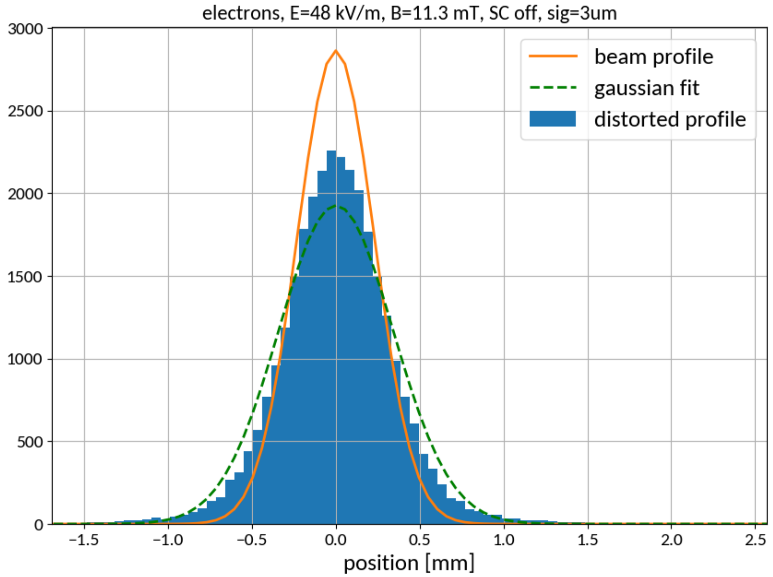
- Some algorithms known since 40’s
  (Gauss Newton or Levenberg-Marquardt).

- 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.

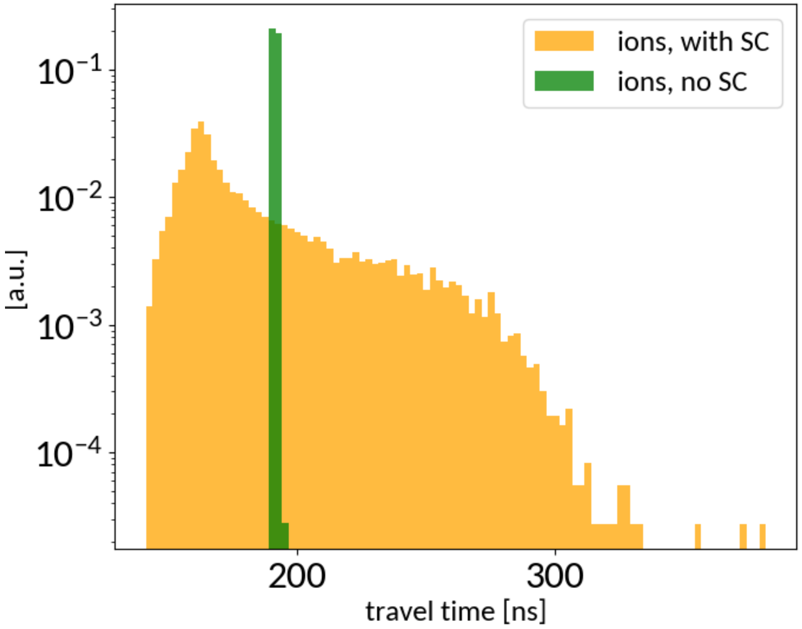
Single-turn magnetic field

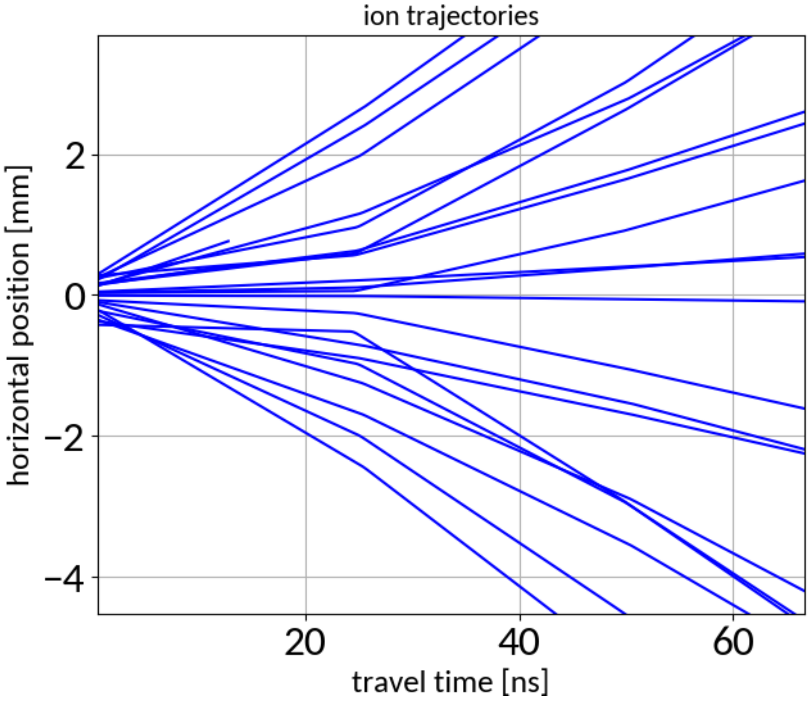
- Influence of vertical beam size





Ion trajectories





MLP Network training (II)

- How it works:
  - Activation function g must be differentiable, eg. sigmoid or tanh.
  - Initial weights chosen randomly.
  - 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)

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

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

  - The cost function gradient is calculated for each layer:

$$\frac{\delta E}{\delta W_{ij}^{1}} = a_j Err_i g'\left(inp_i\right)$$

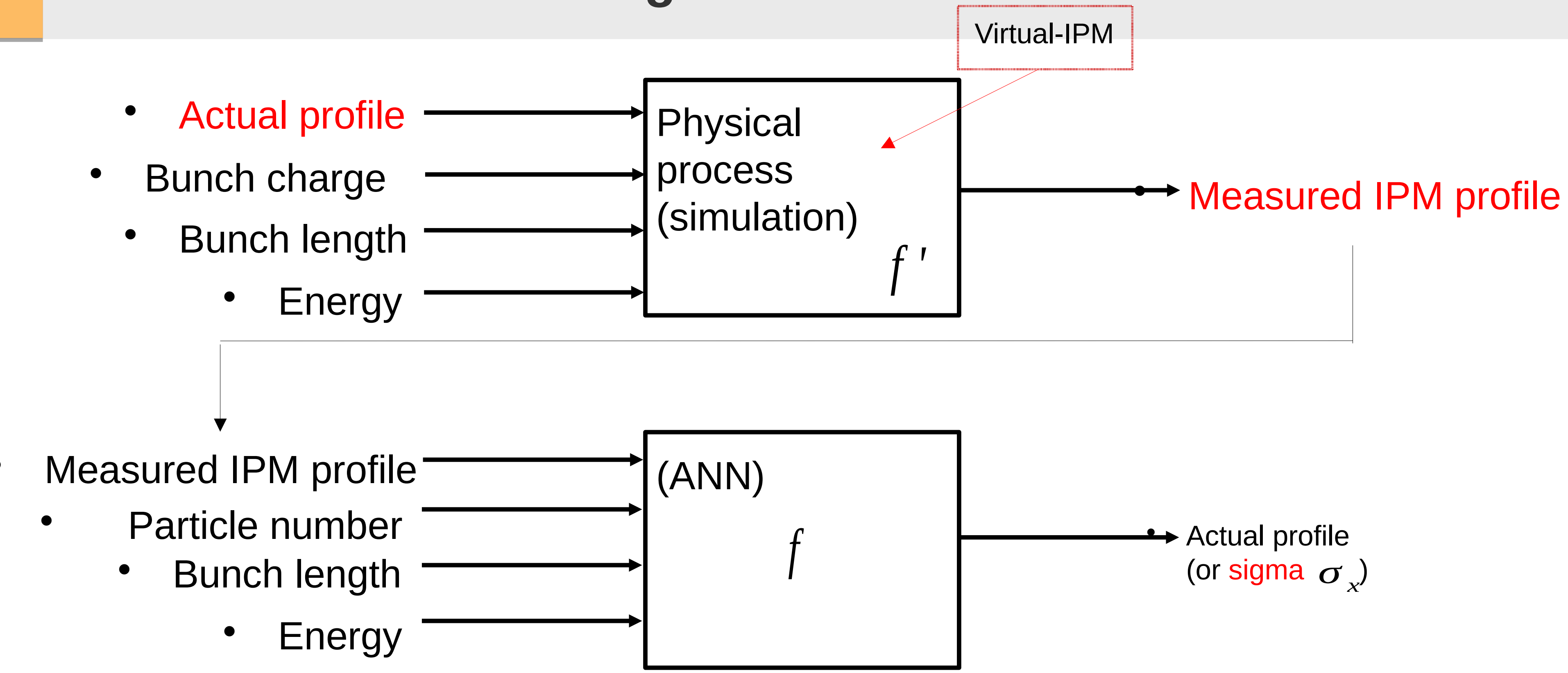
$$\frac{\delta E}{\delta W_{jk}^{2}} = x_k g'\left(inp_j\right) \sum_{j=0}^{M} W_{ij} Err_i g'\left(inp_i\right)$$

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

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

α-learning rate

Profile correction using ANN



Virtual-IPM

- Actual profile
- Bunch charge
- Bunch length
- Energy

Physical process (simulation) f'

Measured IPM profile

- Measured IPM profile
- Particle number
- Bunch length
- Energy

(ANN) f

Actual profile (or sigma σ_x)

Training “grid” (375 points):

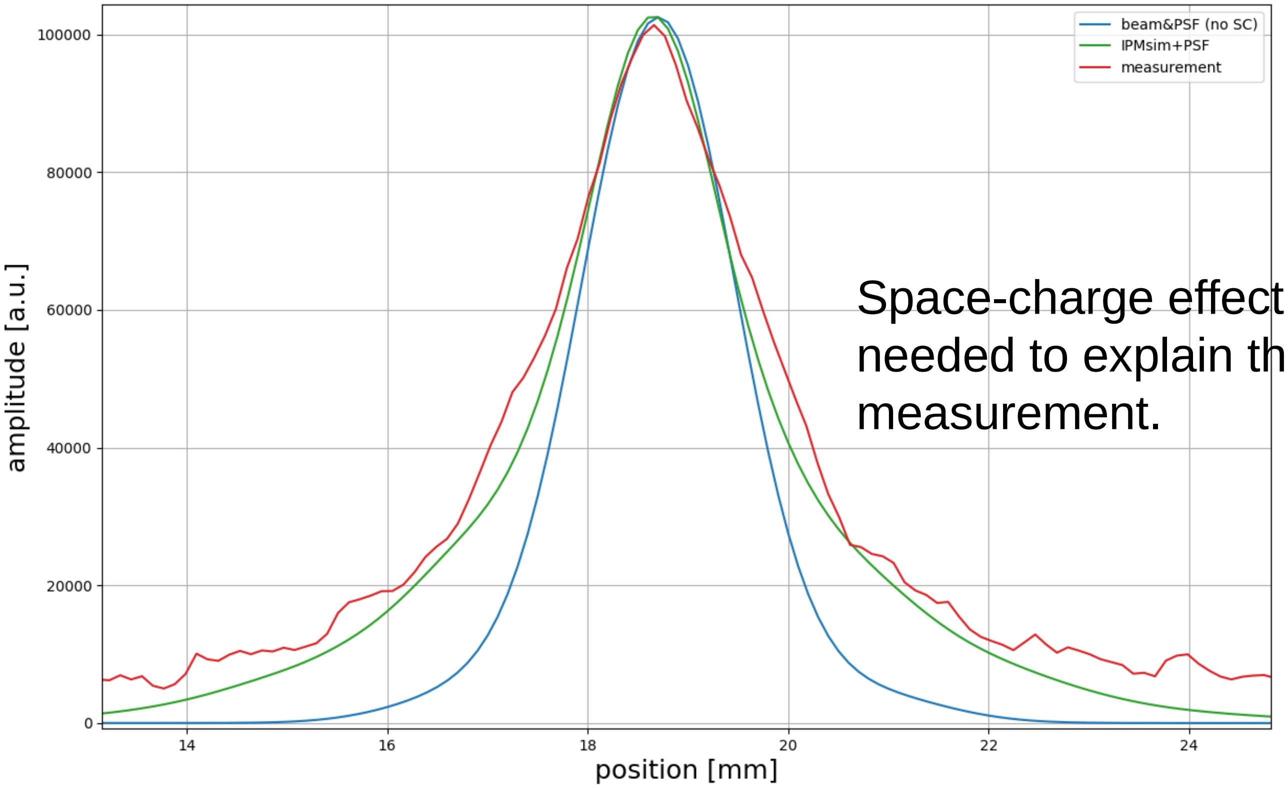
Using *tensorflow* and *Matlab NN toolbox*

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)

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.

Space-charge on SPS beam (16 mT)



Space-charge effect clearly needed to explain this measurement.

MLP Network design (feed-forward)

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.

Results

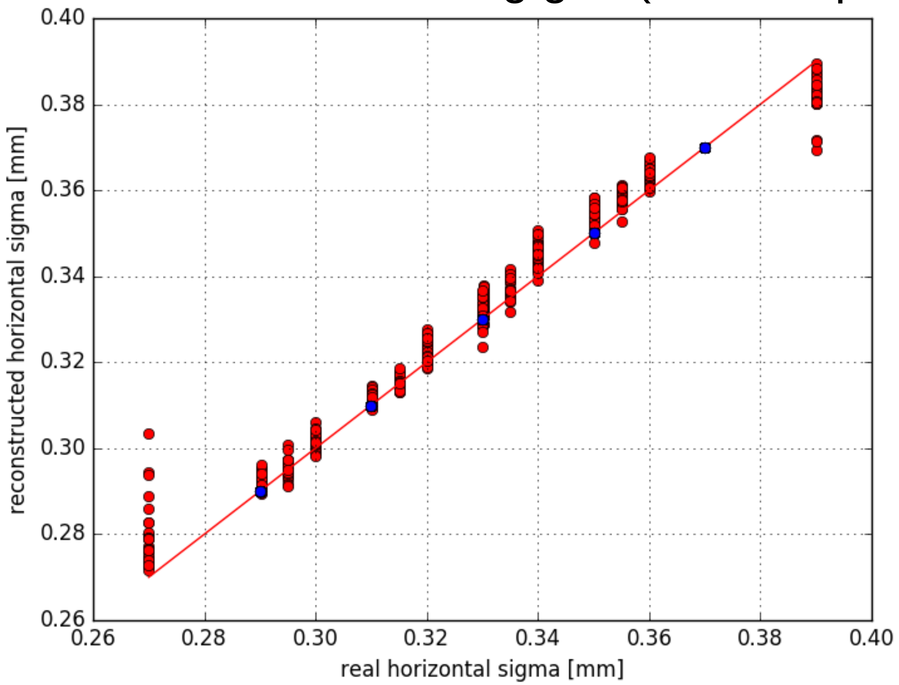
Validation “grid” (128 points)

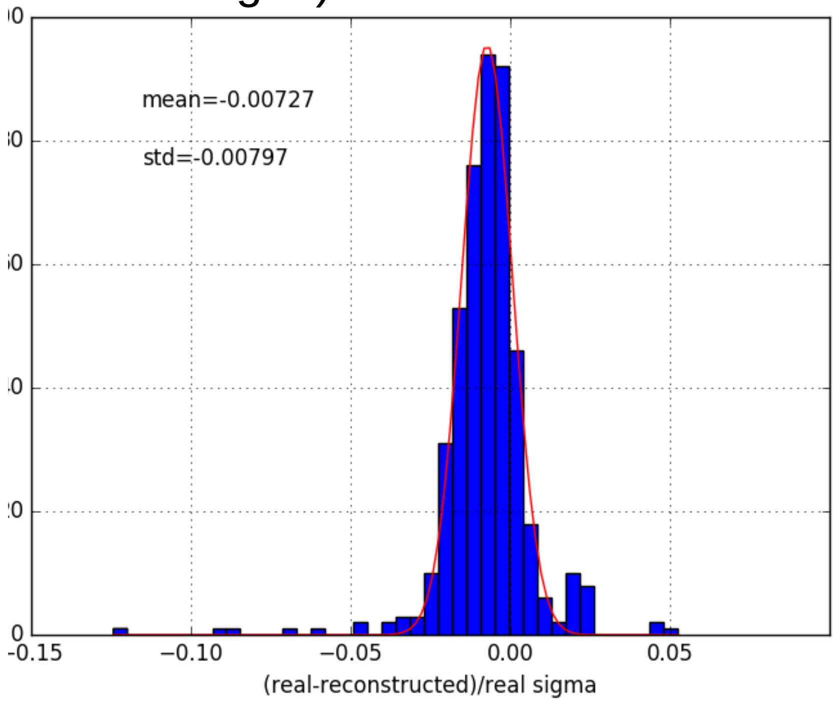
4 validation data sets (inputs and outputs) created:

- 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!

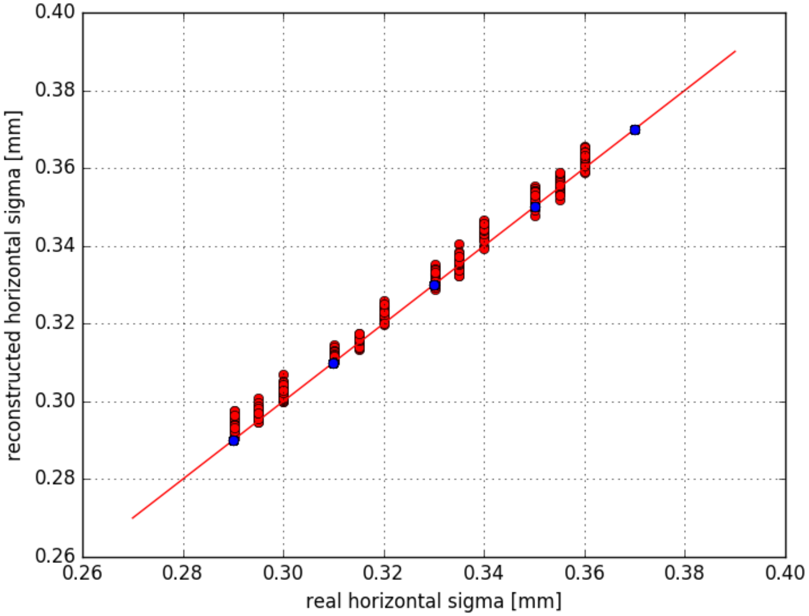


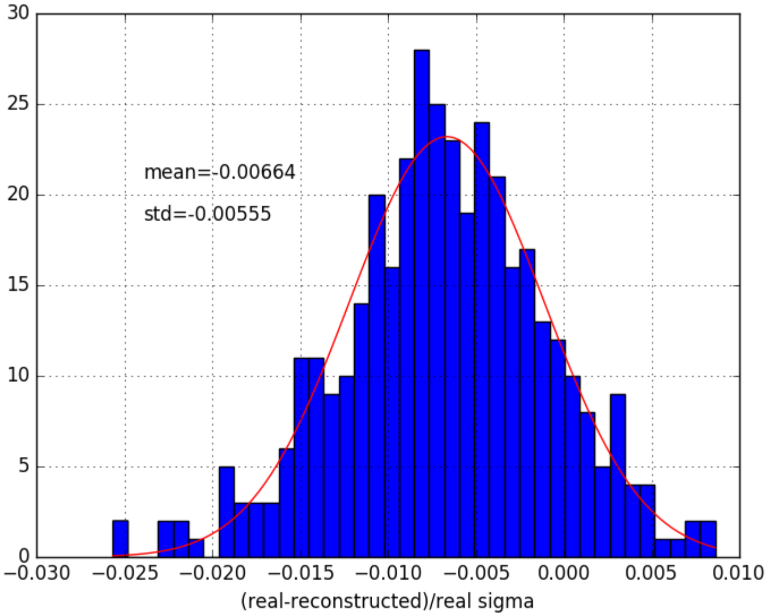


Results

Removing the validation sample outside of “training” area

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





Example 2:
IPM profile corrections

-

- Using higher electric and magnetic fields (expensive, sometimes impractical).

- Electrons + electric and magnetic fields: Sieve method (deconvolve with PSF of radius of Gyration) – difficult in practice.

  [Dominik Vilsmeier, Bachelor Thesis, CERN]

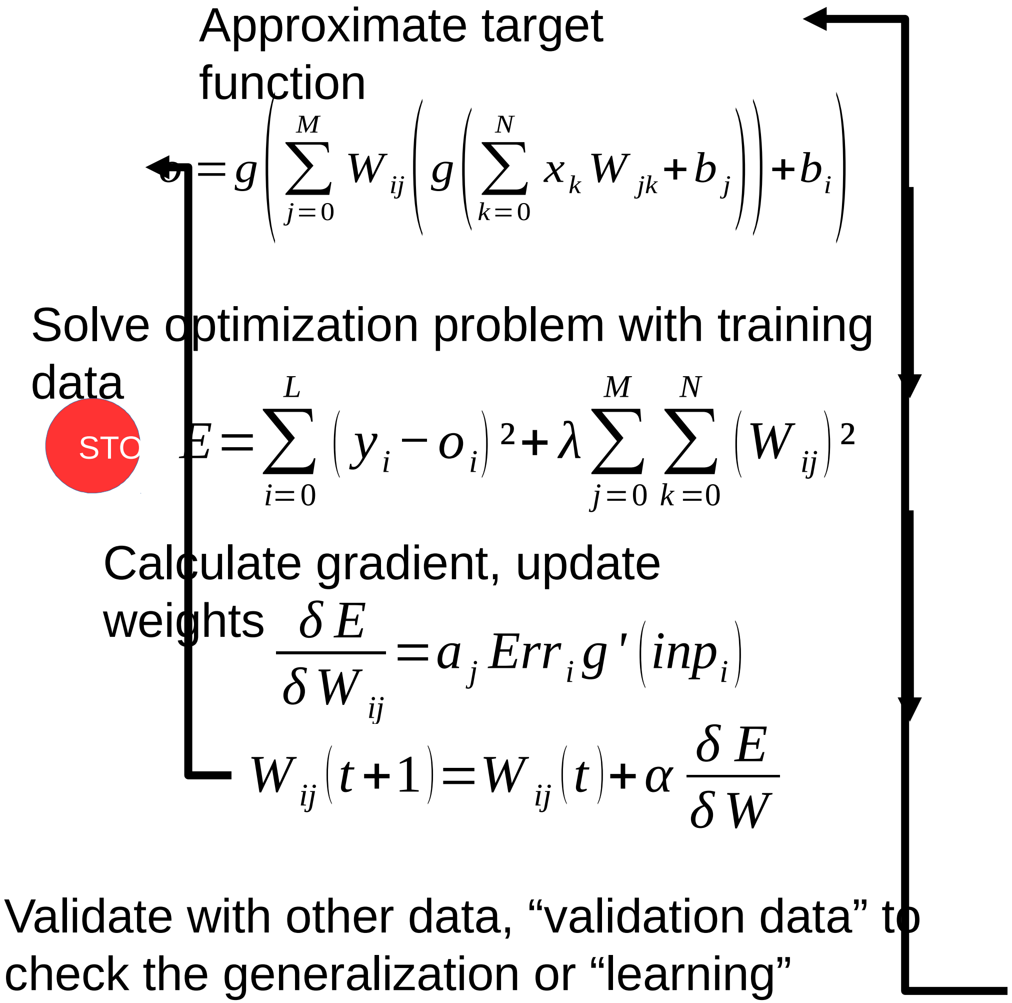
- Electric fields only (ions): several calibration/correction attempts.

  [eg. R. E. Thern, PAC1987, J. Amundson et al., PRSTAB 6, 102801 (2003)]

  Latest work: 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]

Artificial Neural Networks - Overview



Approximate target function

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

Solve optimization problem with training data

STC

$$E=\sum_{i=0}^{L}\left(y_i-o_i\right)^2+\lambda\sum_{j=0}^{M}\sum_{k=0}^{N}\left(W_{ij}\right)^2$$

Calculate gradient, update weights

$$\frac{\delta E}{\delta W_{ij}} = a_j Err_i g'\left(inp_i\right)$$

$$W_{ij}\left(t+1\right)=W_{ij}\left(t\right)+\alpha\frac{\delta E}{\delta W}$$

Validate with other data, “validation data” to check the generalization or “learning”

If not, change the number of units or architecture

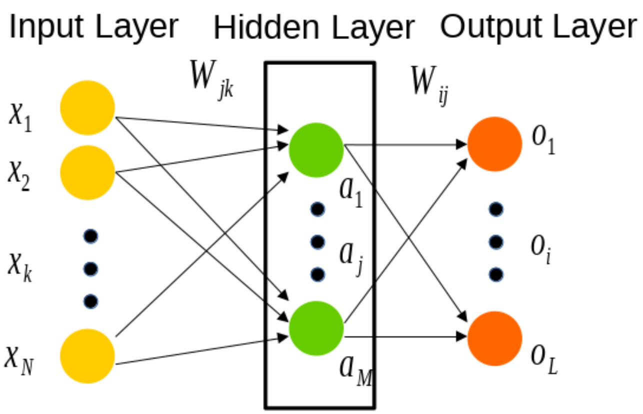
- Supervised learning, unsupervise learning, reinforcement learning
- Batch learning, incremental learn
- Functions: Activation function, Target function, Objective or erro function
- Optimization: Gradient descent, Levenberg-Marquardt, Epoches, Learning rate, Momentum

For more: How could a Kangaroo climb Everest?

- ftp://ftp.sas.com/pub/neural/kangaroos

- Generalization: Cross validation, regularization, early stopping

Calculating backpropagation



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

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

$$\frac{dE}{dW_{11}} = \frac{dE}{do} \cdot \frac{do}{ds} \cdot \frac{ds}{dW_{11}}$$

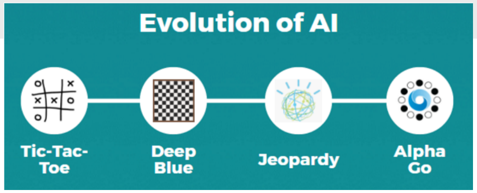
-2(y-o)

g'(s)

a_1

$$\frac{\delta E}{\delta W^1_{ij}} = a_j Err_i g'\left(inp_i\right)$$

Examples of ML-based projects (I)



- AlphaGo :
  - Go is difficult for algorithms because of number of configurations (>2x10^{170},chess only ~5x10^{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.