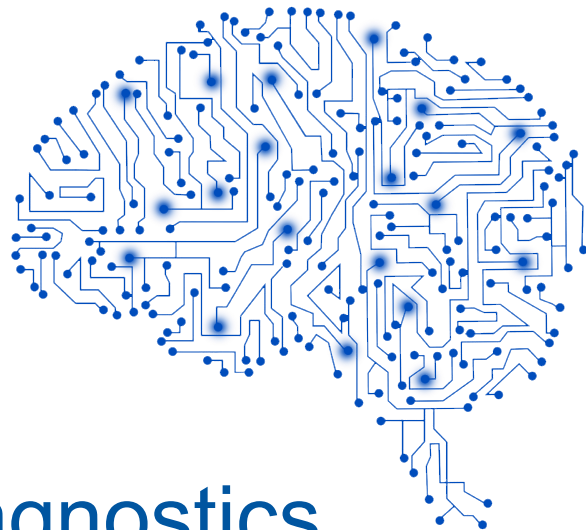






Tutorial on Application of Machine Learning to Beam Diagnostics

Elena Fol
CERN, Goethe-University Frankfurt

7th International Beam Instrumentation Conference, 9 – 13 September 2018, Shanghai

Outline

- I. Introduction to Machine Learning
 - General application fields
 - Relevant concepts and definitions
- II. Application to Beam Diagnostics
 - Motivation
 - Examples of recent applications
 - Potential further applications
- III. Conclusion and recommendations

Part I. Introduction to Machine Learning

Teaching machines to learn from experience

- Tasks that are extremely easy and obvious for us are difficult to program in traditional ways
- Impossible to learn every possible rule to perform a task
 - learn from examples instead

Teaching machines to learn from experience

- Tasks that are extremely easy and obvious for us are difficult to program in traditional ways
- Impossible to learn every possible rule to perform a task
 - learn from examples instead



Teaching machines to learn from experience

- Tasks that are extremely easy and obvious for us are difficult to program in traditional ways
- Impossible to learn every possible rule to perform a task
 - learn from examples instead



→ Cat?

Machine Learning is extremely successful in many different fields:

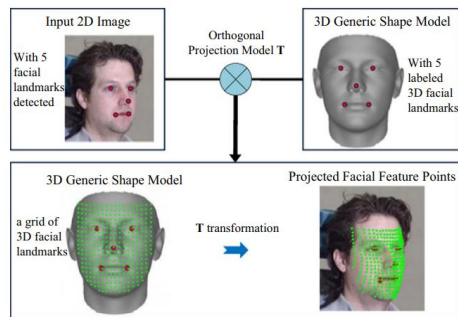
- Computer vision
- Speech recognition
- Natural language and text processing
- Face recognition
- Financial market analysis, risk prediction
- Search engines
- Medical diagnostics
- Transactions fraud detection
- Recommendation engines, advertising
- Robotics, automation
- Video games
- Self-driving cars

MNIST handwritten digits dataset

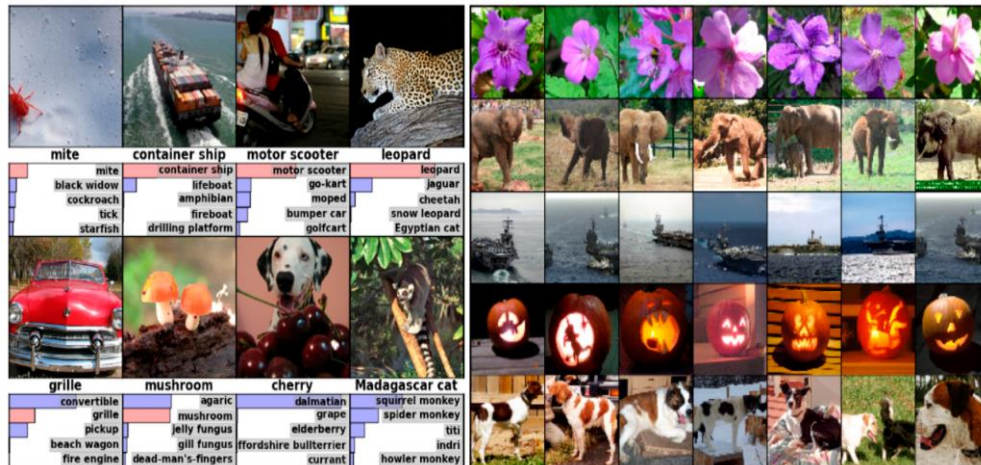


<http://yann.lecun.com/exdb/mnist/>

Y. LeCun, et.al, "Gradient-based learning applied to document recognition"



D. Changxing, T. Dasheg, "Pose-invariant face recognition with homography-based normalization"



The ImageNet project

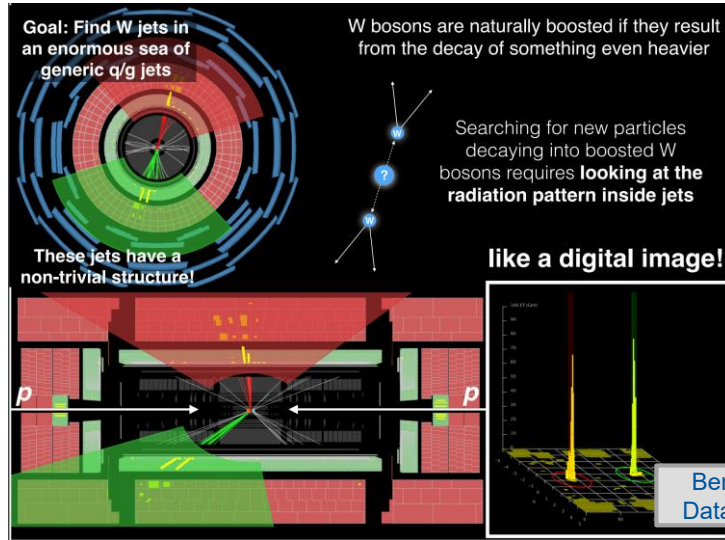
- Visual objects recognition (up to 78% accuracy on 1000 object classes)
- G. E. Hinton et.al, "ImageNet Classification with Deep Convolutional Neural Networks"

Face recognition and reconstruction

- Automatic detection of semantic regions
- Specific "layers" are sensitive to certain regions (e.g. eyes, nose, lips)

AlphaGo from Google

- First match against Go European champion in 2015, 5:0 for AlphaGo
- In 2017 AlphaGo surpassed the performance of its previous versions and became the strongest Go player of all time *



High Energy Physics

- ML is used in dark matter search, jets recognition, particle tracking, neutrino classification, shower simulations

* Silver, David et al. "Mastering the game of Go without human knowledge." *Nature* 550 (2017): 354-359.

Relevant ML concepts and definitions

*"... computer programs and algorithms that automatically improve with experience by learning from examples with respect to some class of task and performance measure, without being explicitly programmed." **

Supervised Learning

- Input/output pairs available
- Make prediction for unknown input based on experience from given examples

Automatic spam detection,
object detection in computer
vision, speech recognition,
predictive control

Unsupervised Learning

- Only input data is known
- Learn structures and patterns

Anomaly detection, pattern
recognition, clustering,
dimensionality reduction

Reinforcement Learning

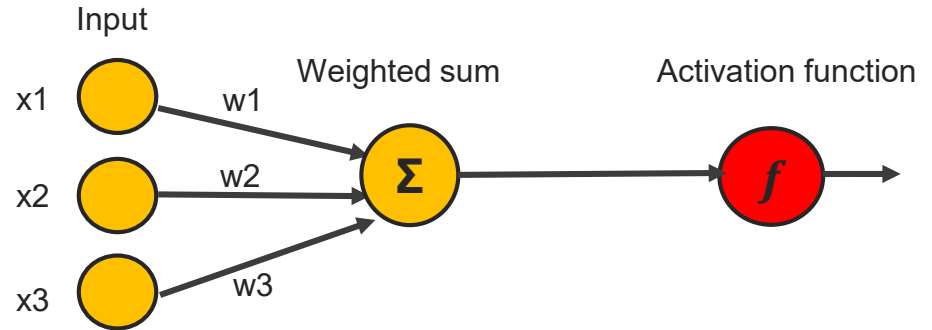
- No training data
- Interact with an environment
- Trying to learn optimal sequences of decisions

Robotics, industrial
automation, dialog systems

* Thomas M. Mitchell. Machine Learning. McGraw-Hill, Inc., New York, 1997.

Artificial Neural Networks

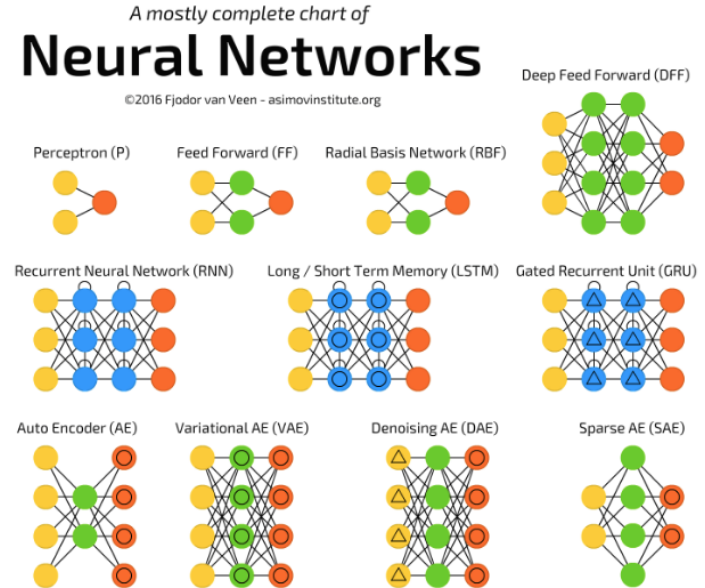
- Basic processing unit = **neuron** (or perceptron) with following parameters:
 - **Weights** w from the inputs x_i
 - **Activation function** f
 - Output y of a single neuron: $y = f(\sum x_i w + b)$
- Neurons are stacked into **layers**
- Connected layers build a **network**



Artificial Neural Networks

Many different architectures:

- Hidden layers increase the complexity of the network and allow to solve **non-linear problems**
- "Deep Learning"
- **Architecture heavily depends on particular problem**



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

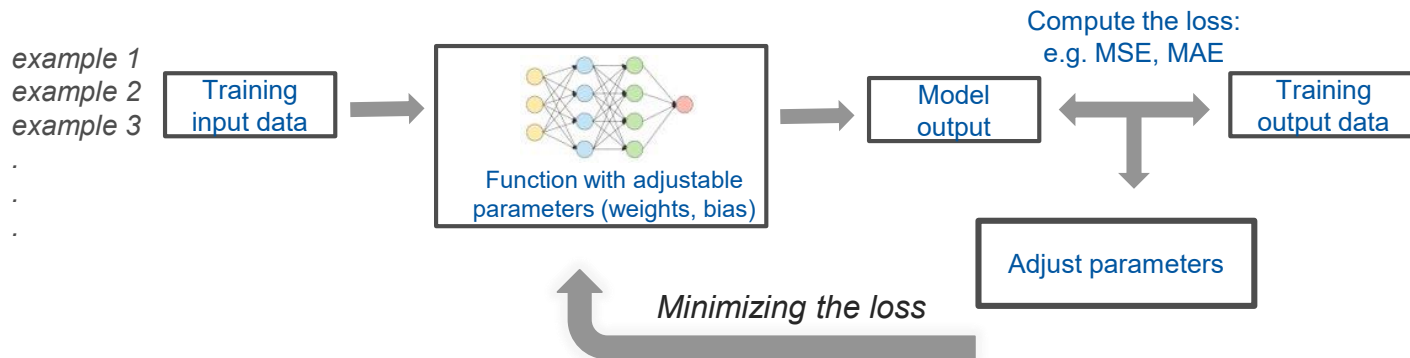
Artificial Neural Networks

- **Universal Approximation Theorem**

A simple neural network including only a single hidden layer can approximate any bounded continuous target function with arbitrary small error.

(Cybenko, 1989, for sigmoid activation functions)

- **How does the learning work in practice?**



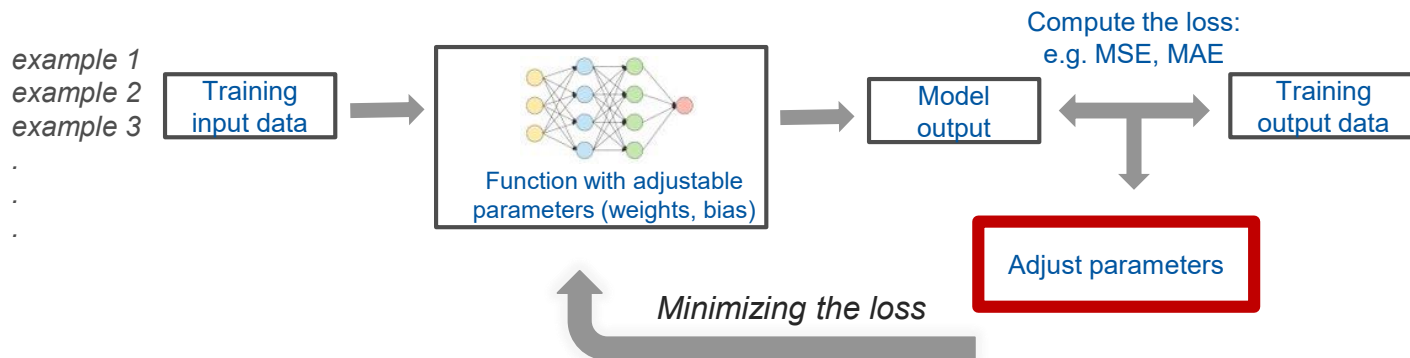
Artificial Neural Networks

- **Universal Approximation Theorem**

A simple neural network including only a single hidden layer can approximate any bounded continuous target function with arbitrary small error.

(Cybenko, 1989, for sigmoid activation functions)

- **How does the learning work in practice?**



Backpropagation with Gradient Descent

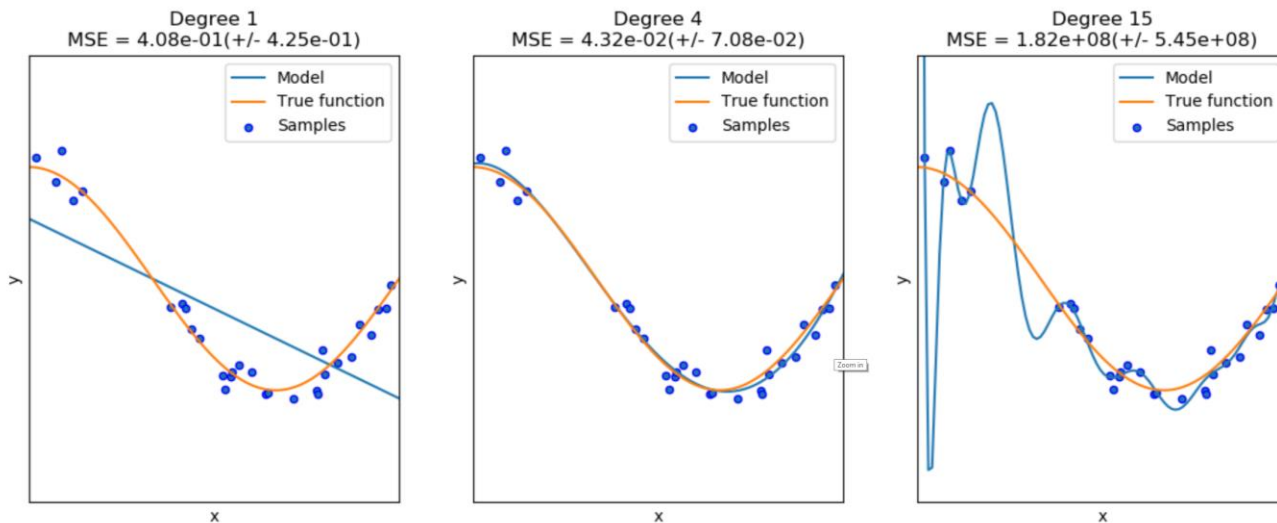
Artificial Neural Networks: Backpropagation and Gradient Descent optimization

- How to minimize **loss** L using **weights** w ?
 - **Gradient Descent** and its improvements (Stochastic Gradient Descent, AdaGrad, Adam)
- Forward step: compute and save intermediate computations
- Final loss is composed by output of nonlinear hidden layers
- Backward step:
 - For each layer, compute gradient of loss w.r.t. parameters
 - Update parameters $w_{t+1} = w(t) + \alpha \frac{\delta E}{\delta w}$

learning rate α to control the size of the gradient step

- Repeat until parameters are stable or desired loss is achieved: **when to stop the training?**
 - **Validation set is needed to determine when to stop the training to obtain an optimal model**

Training and generalization: no perfect model needed!



Simple models underfit

- Derivate from data (high bias)
- Do not correspond to data structure (low variance)

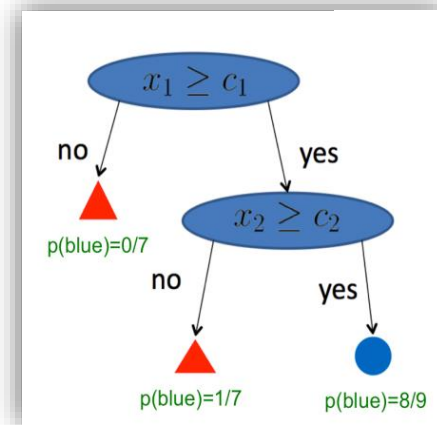
Bias-Variance tradeoff

Complex models overfit

- Very low systematical deviation (low bias)
- Very sensitive to data (high variance)

Decision Trees

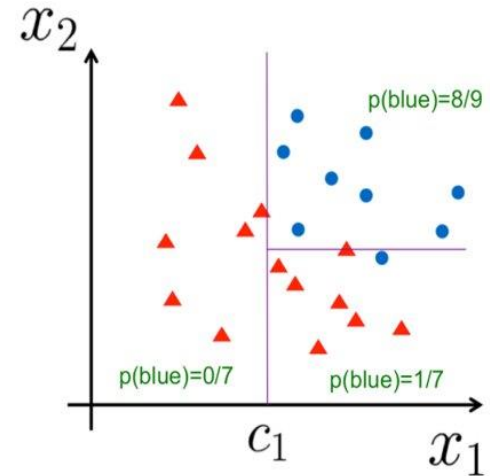
- Split the input data based on a sequence of variables (thresholds) to estimate the target output value
 - threshold selection depends on the chosen algorithm
- Leaves specify the predicted value of the target output variable
- Choose parameter and threshold for splitting aiming to minimize the difference between the value in the leaf and true value
 - e.g. Mean Square Error for regression
 - Cross-entropy for classification



M. Kagan, CERN Academic
Training Lectures

Decision Trees

- After the tree is constructed, prediction can be made by following the edges until one of the leaf nodes is reached
- Simple to understand and visualise (depending on the tree depth)
- Suitable for regression and classification



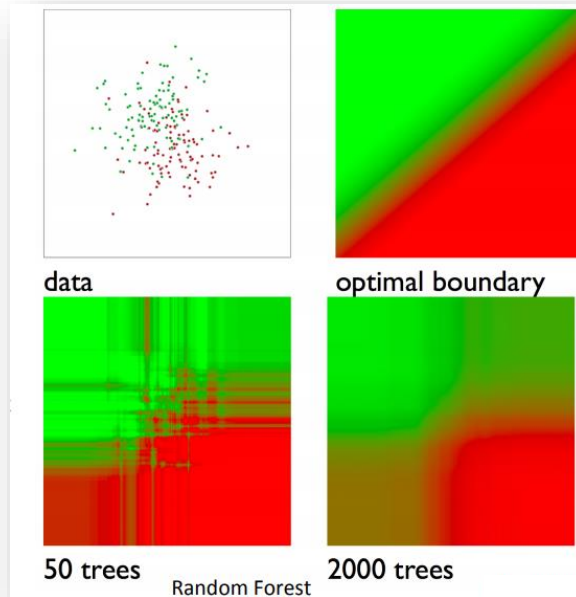
Ensemble methods

Single trees tend to overfit data and can be unstable to small variations in data

- **Ensemble methods:** Train several slightly different models and take majority vote/ average of the prediction

Random Forest is one of the most commonly used algorithms

- Selects random subset of examples, train separate model on each subset
- Only random subset of features is used at each split
- Increase variance, tend not to overfit



M. Kagan, CERN Academic
Training Lectures

Clustering: Unsupervised Learning

No labeled data is needed

Grouping or separating data objects into clusters

- Objects within a cluster are more similar than to objects from other clusters
- Similarity = distance metrics, density of objects inside a cluster, e.g. *euclidean distance* for two data objects p and q

$$d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

- Summarized representation of the data
- Can find hidden patterns in the data, similarities and differences

Clustering

Simplest clustering algorithm: K-means*

- Starting with randomly chosen data points as cluster centers, each point is assigned to the closest center
- Move centers to the centroids of the clusters, reassign the points to the nearest center
- Repeat until moving the centroid gives no improvement (based on total squared distance between each point and cluster centroid)



*Stuart P. Lloyd. Least squares quantization in PCM

Useful resources for further introduction:

- Elements of Statistical Learning, Hastie, Tibshirani, Friedman (2009)
- Pattern Recognition and Machine Learning, Bishop (2006)
- Stanford Course on Machine Learning: Ng <https://cs229.stanford.edu/>
- M. Kagan, Academic Lectures at CERN:
<https://indico.cern.ch/event/619370>

Part II. Application to Beam Diagnostics

Motivation

Beam Diagnostics

Limitations of traditional
optimization and modeling
tools?



ML is a powerful
tool for prediction and data
analysis

**Which limitations can be solved by ML
with reasonable effort?**

Some traditional optimization methods: Newton's method, Simplex, Random walk optimization

- Resolve **linear** correlations between input parameters and optimization objectives
- Relatively **small** amount of target parameters

Limitations:

- How to deal with **non-linear** behavior?
- Required computational resources for **large** amount of optimization targets
- Objective functions, specific rules and thresholds **have to be known**

Machine Learning methods can learn an arbitrary model from given examples without requiring explicit rules

Potential for Machine Learning in Beam Diagnostics

Prediction and
optimization of
beam parameters

Automation of
diagnostics and
operation

Beam control and
lattice imperfection
corrections

Detection of
instrumentation
defects

Optimization and prediction of beam parameters

Predict parameters that are obtained by complex and slow diagnostics

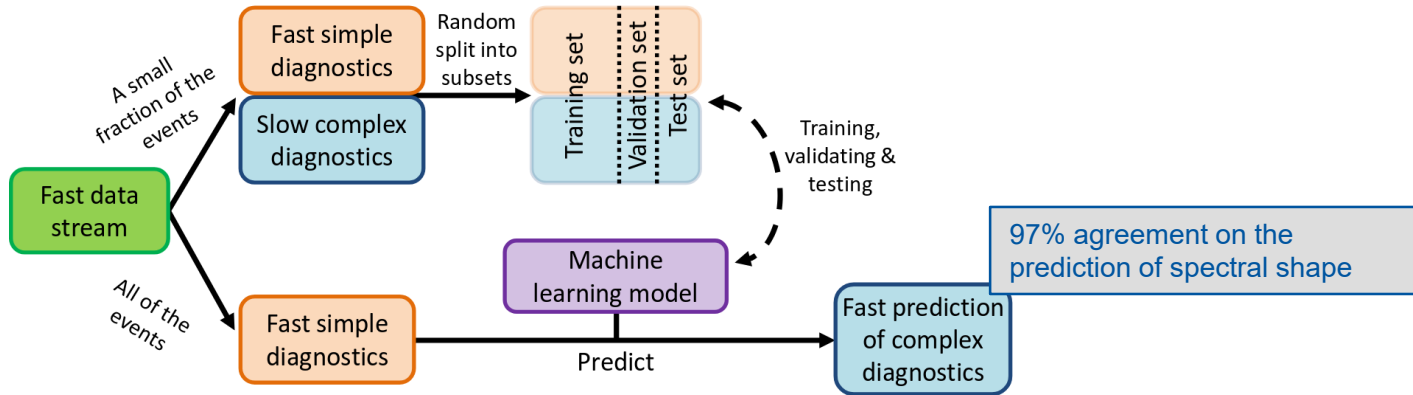
Machine learning applied to single-shot x-ray diagnostics in an XFEL,

A. Sanchez-Gonzalez, et al. <https://arxiv.org/abs/1610.03378>

- Input: simple electron beam and x-ray parameter
- Output: photon energy, delay between two x-ray pulses, spectral shape
- Physical process behind correlations between input and output, but modelling of every experimental aspect is not possible

Optimization and prediction of beam parameters

- More than 300 input variables initially, using correlation analysis and PCA reduced to 40
- Linear regression, Support Vector Regression, ANN
- Method is applicable at any XFEL facility



More similar approaches for beam control and tuning systems using ANN:

A. L. Edelen, S. G. Biedron, B. E. Chase, D. Edstrom, S. V. Milton, P. Stabile, *Neural Networks for Modeling and Control of Particle Accelerators*, *IEEE Transaction of Nuclear Science*, 63 (2), 2016

Optimization and prediction of beam parameters

Mapping between distorted IPM profile and original one

R. Singh, M. S., D. Vilsmeier, Simulation supported profile reconstruction with Machine Learning, Proc. of IBIC17 (WEPPC06)

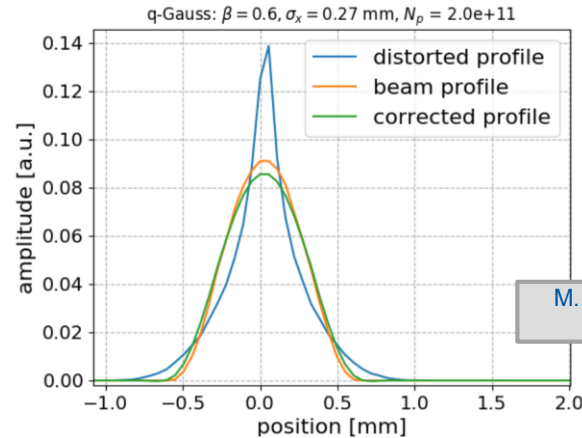
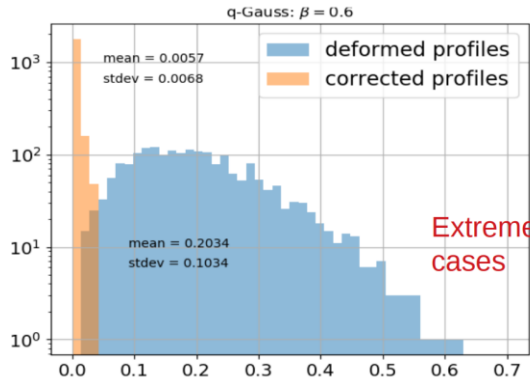
D. Vilsmeier et al., Reconstructing Space-Charge Distorted IPM Profiles with Machine Learning Algorithms, Proc. of IPAC 2018 (WEPAC008)

- IIPM profile distortion is well studied, but the problem is too complex to use algorithms for profile correction
- Estimate the actual beam profile width from measured profile distorted by space charge
- Input: simulated distorted profiles
- Output: profile without distortion

Optimization and prediction of beam parameters

- Methods used: Linear regression, SVR, ANN
- Already Linear regression gives very good results, best results achieved with ANN

Neural network trained only on gaussian profiles!



M. Sapinski, HB2018,
Daejeon Korea

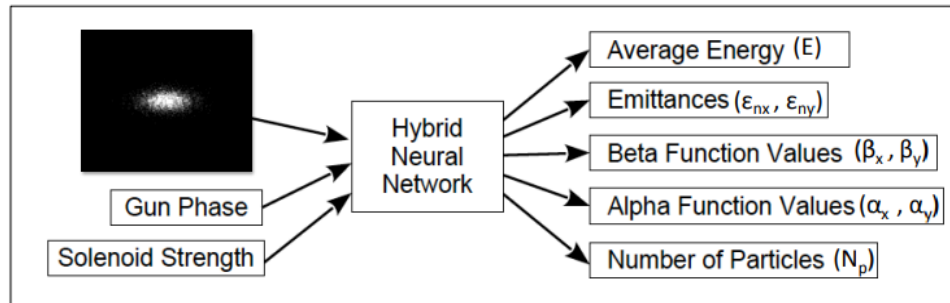
- ANN learns about nature of space-charge deformation, not just about transformation of gaussian profile

Optimization and prediction of beam parameters

Image-based prediction of multiple beam parameters

"First steps toward incorporating image based diagnostics into particle accelerator control systems using Convolutional Neural Network", A.L. Edelen et al. NAPAC16 (TUPOA51)

- Convolutional Neural Networks (CNN) are very successful in image recognition
- CNN and fully-connected ANN are used to incorporate image-based and non-image-based data into the model to predict multiple beam parameters via



Neural network inputs and outputs.

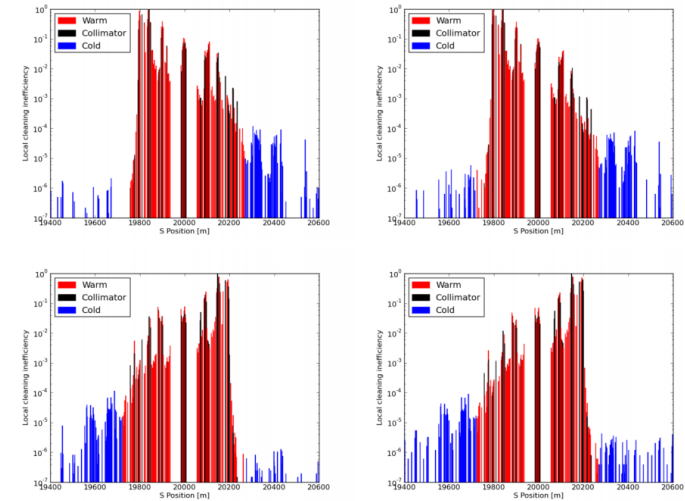
mean absolute errors are between 0.4% and 3.1% of the parameter ranges

Automation of operation and diagnostics

Exploring characteristics of beam loss patterns

Machine Learning applied at the LHC for beam loss pattern classification, G. Valentino, B. Salvachua, IPAC18 (WEPAF078)

- Determining losses characteristics helps to understand the impact on luminosity and lifetime of accelerator components
- Input: Losses measured at BLMs
- Output: one of 4 types of beam losses (classification)
- Gradient Boosting Classifier*
- Applied during the beam squeeze in the LHC machine cycle



Classification success rates between 95% and 100% have been achieved

*J. H. Friedman, "Greedy function approximation: A gradient boosting machine". Ann. Statist. 29 (2001), no. 5, pp. 1189- 1232

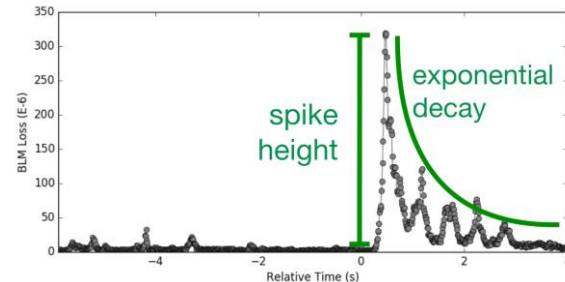
Automation of operation and diagnostics

Automatic alignment of collimators

"Automatic angular alignment of the LHC collimators", G. Azzopardi et al., ICALEPCS17

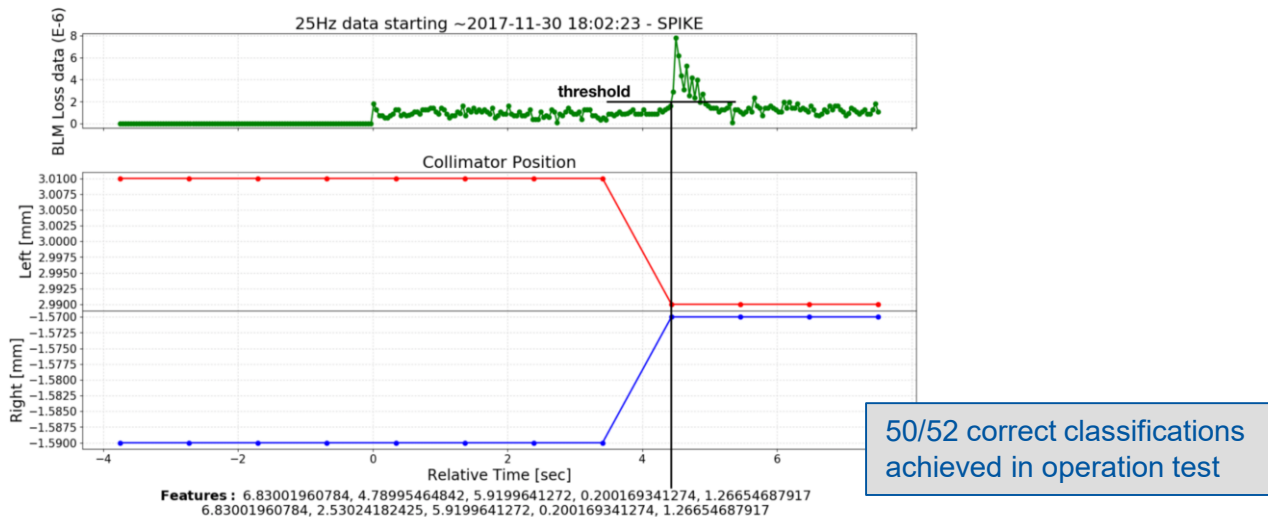
"Spike Pattern Recognition for Automatic Collimation Alignment", G. Azzopardi, No. CERN-ACC-NOTE-2018-0010

- Collimators have to be realigned during operation due to orbit shifts and beam parameter changes
- If loss spike is above a pre-selected threshold, the collimator is stopped: requires an expert to determine if the collimator actually has touched the beam
- Input: spike height, exponential decay and collimator jaw position
- Output: if collimator is aligned or not (classification)



Automation of operation and diagnostics

- Ensemble of several ML methods used – use the majority vote of all models
- Tested at the SPS and the LHC



TCTPH.4R2.B2 both jaws spike

Beam control and lattice imperfection correction

Opportunities to build beam diagnostics and control systems using ML has been studied already since early 90's

- Orbit corrections studies:
 - E. Bozoki, "Neural Network technique for orbit correction in accelerators", 1994
 - Y. Kijima, "A beam diagnostic system for accelerator using Neural Networks", 1992
 - E. Meier, "Orbit correction studies using Neural Networks", 2012

Beam control and lattice imperfection correction

- Lattice imperfections corrections: Y. Kijima, "A beam diagnostic system for accelerator using Neural Networks", 1992
 - Detect dipole errors aiming to develop rapid commissioning
 - Obtain dipole errors from measured beam position
 - Small machine (8 BPMs, 8 fodo-cells)
 - In the presence of more than 2 dipole field effects, the model performance decreased significantly
- *Anyway, obtained results have shown first potential of ML to be applied on optics corrections*

Prediction of correction knobs settings at LHC

- Optics correction: identify quadrupole strength changes needed to minimize the deviation from nominal model
- Find knob settings = multivariate regression
- Simulations dataset, 100 000 samples
- Input: 1046 phase errors per beam (measured at 1046 BPMs in both planes)
- Output: 190 correction variables
- Several ML model have been applied, best performance on the test set achieved by **Random Forest (explained variance: 0.99, MAE 0.02×10^{-5})**

1. Build data set

Simulate measurements by introducing random errors to nominal model

- 1024 input parameters
- 190 outputs

2. Training

Input: Phase advance deviation from simulated measurements

Take a **subset** of simulated data in order to train the model



Target output: introduced errors

Errors predicted by the model



Errors introduced to simulate measurements

Split minimizing impurity

Impurity measure:
e.g. Mean Squared Error

3. Validation and Test

- Validation set to **tune model parameters**
- Test set to **measure model performance**

Most important parameter to tune by Random Forest:

- Number of trees
- Minimal number of samples in one split
- Maximum depth of a tree

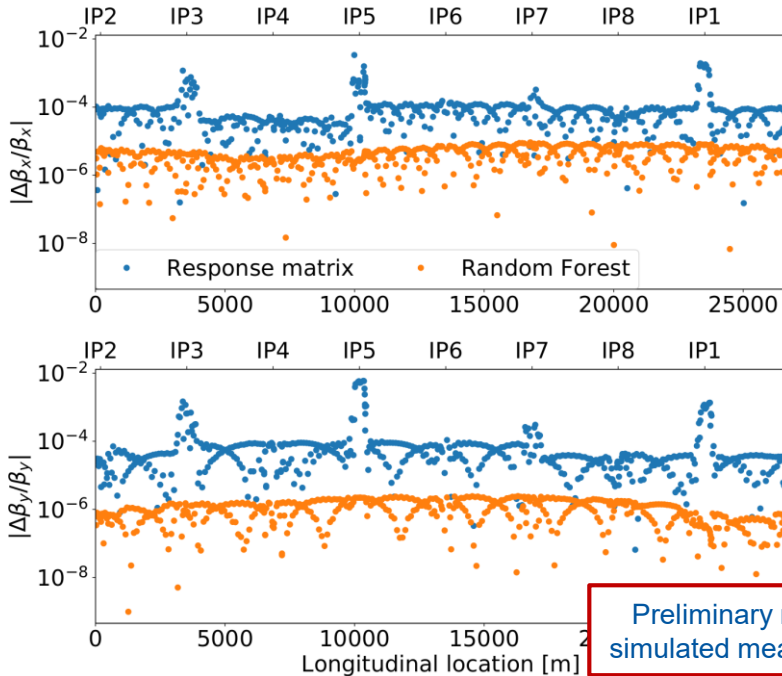
4. Prediction

Obtain the corrections based on a **new** measurement



- *Generalized model of your data*
- *Prediction can be made based on new input since the model "knows" the correlations between input and output*

Comparison to traditional Response Matrix approach, expected β -beating after applying corrections:



Preliminary result on simulated measurement

Random forest outperforms traditional correction method

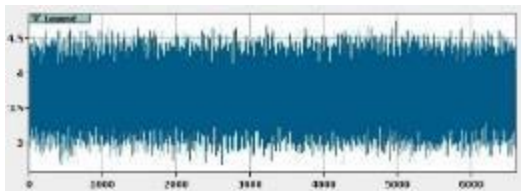
Further work:

- Include other sources of errors and BPM noise into simulations
- Train model on different optics settings to achieve better generalization
- Add real measurements and corresponding corrections to the training set
- Try other ML models

Anomaly detection: Detection of faulty BPMs

Optics measurements at LHC

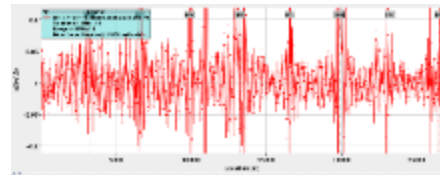
BPMs record the turn-by-turn data measuring the oscillations of the excited beam



Calculate optics functions (beta-beating, dispersion, etc.) based on harmonic analysis of BPMs signal



+ data cleaning



- Unphysical values coming from faulty BPMs signal still can be observed in reconstructed optics even after cleaning with available tools
 - **ML as an alternative solution to improve the analysis**

Anomaly detection: Detection of faulty BPMs

- 1024 BPMs per beam around the ring to measure turn-by-turn data
Statistical analysis of the past measurements shows that **~10%** of BPMs are faulty

General Idea

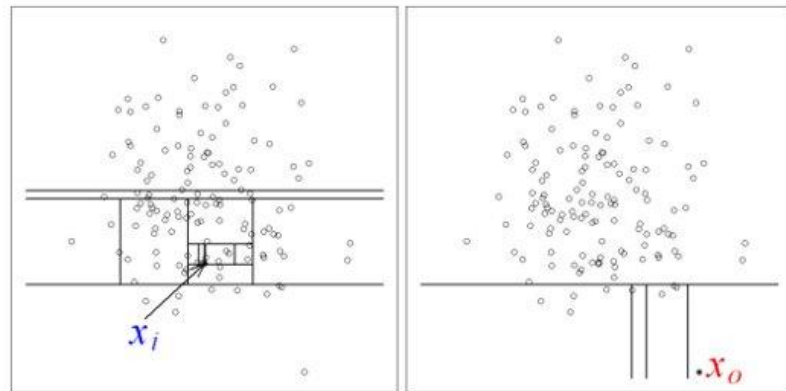
- We do not want to replicate current results, no training data set available: **Unsupervised learning approach**
- Assuming most of the BPMs measure correctly, the bad BPMs should appear as an **anomaly**
- Consider combination of different parameters, separate the data
- 3 parameters: Tune, Amplitude, Noise
- Applied algorithms: K-means[1], DBSCAN[2], Local Outlier Factor[3], **Isolation Forest**[4]

1. Stuart P. Lloyd. Least squares quantization in PCM
2. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise" Ester, M., H. P. Kriegel, J. Sander, and X. X.

3. Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000, May).
LOF: identifying density-based local outliers
4. Liu, Fei Tony, Ting, Kai Ming and Zhou, Zhi-Hua. "Isolation forest." Data Mining, 2008. ICDM'08.

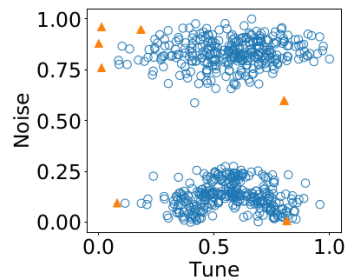
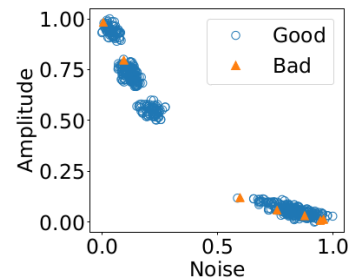
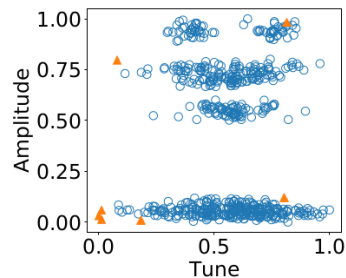
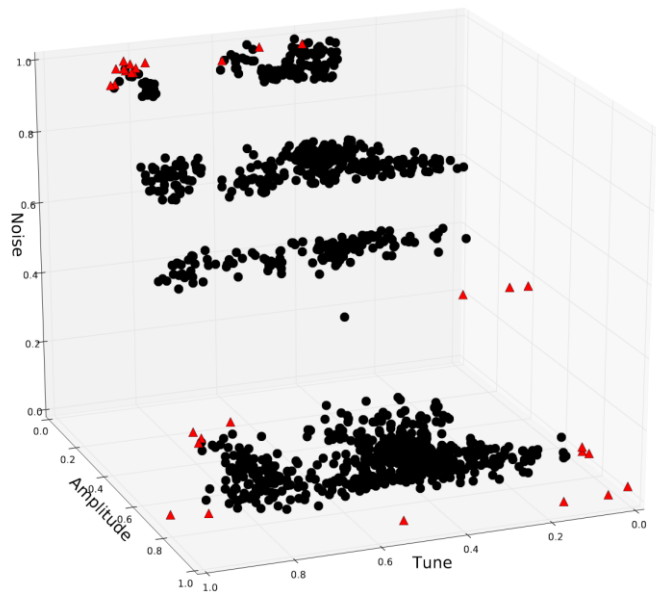
Isolation Forest

- Randomly selects a parameter and then randomly selects a split between minimum and maximum values of the selected parameter.
- Random tree: the number of splits required to isolate a data point is equivalent to the path length from the root to the terminating node.
- Forest: Many random trees perform the splitting – path length, averaged over the forest is a score of "normality".
- Shorter paths are produced for anomalies.

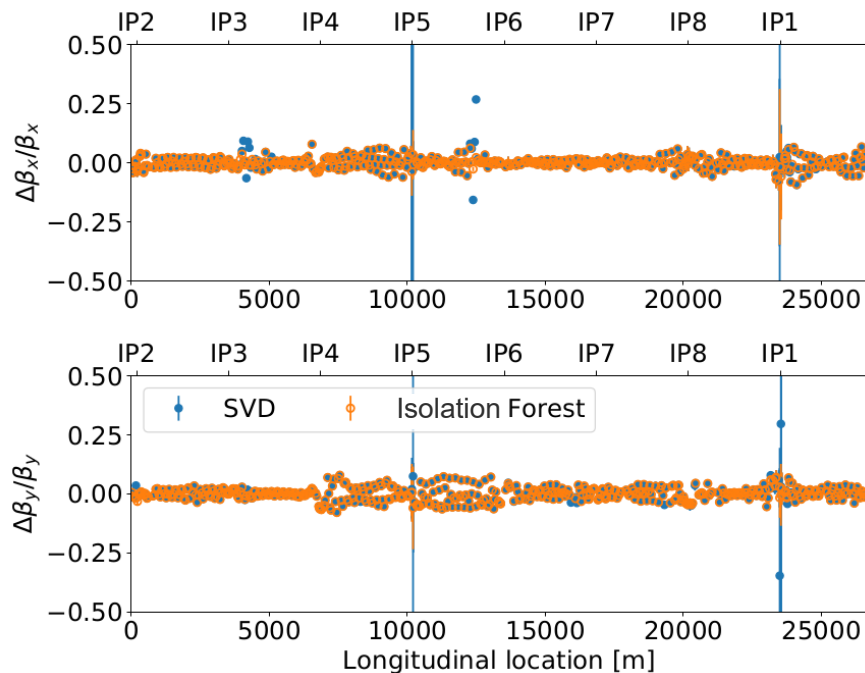


IF cluster analysis on arcs measurements of tune, amplitude and signal noise in horizontal plane and its 2D-projection:

The data is normalized to the range $[0,1]$ and separated into IR and Arcs BPMs due to the different data points distribution in these regions



β -beating from the measurement cleaned with SVD before and after applying IF:

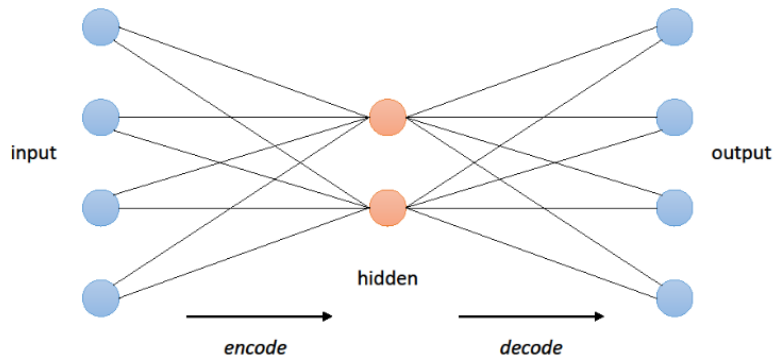


- This method is fully integrated into optics measurements at LHC
- Successfully used during beam commissioning and machine developments in 2018 under different optics settings

Possible alternative: Autoencoder

How does an autoencoder work?

- Neural Network with specific structure
- Tries to reproduce in its output whatever comes in the input: $f(x) = x$
- Encoder: compressing the input data to lower dimensions
- Decoder: Reconstructing the data into original input



General idea for faulty BPMs detection:

- Since the majority of BPMs are good, the trained model will learn the behavior of valid signal
- The loss function will measure the difference between a given point and the learned general case
- Loss above a defined threshold -> anomaly detected!

Potential further applications

- Maintenance of accelerator systems
- Dynamic aperture computation avoiding costly simulations
- Reduction of the complexity of current analysis methods using feature importance and dimensionality reduction methods (e.g. autoencoders and decision trees)
 - Example: Beam lifetime optimization at the LHC,
<https://indico.cern.ch/event/738306/>

Part III. Recommendations and conclusion

Where can we use ML in Beam Diagnostics?

- Simultaneous optimization of several beam parameters
- Prediction of beam behavior
- Automation of diagnostics and operation
- Lattice imperfection correction
- Detection of instrumentation defects

... more great ideas are welcome during discussions!

Practical advice

- Often data preprocessing is needed before any model can be applied: rescaling, feature engineering, denoising, outlier elimination:
 - data visualization can help
- Start with simple models - increase complexity only if needed
- Estimate model generalization (75% train, 15% validation, 10% test)
- Frameworks to use:
 - Prototyping, fast and easy implementation (very good documentation): <http://scikit-learn.org/>
 - High-level package for Neural Networks: – <https://keras.io/>
 - Deep Learning, specific complex model architectures:
<https://www.tensorflow.org/>
<http://deeplearning.net/software/theano/>

Conclusions

- **Important to identify where ML can surpass traditional methods**
 - How much effort is needed to implement a ML solution? Is appropriate infrastructure for data acquisition available? Enough resources to perform the training?
- **Good examples**
 - Automation of particular accelerator components e.g. collimation system
 - Modeling is not possible, fast diagnostics needed, e.g. beam profile reconstruction, image-based diagnostics
 - When training data is already available, e.g. optics measurement and correction, beam loss maps
 - Unsupervised learning to apply directly on the data, to discover important correlations, e.g. detection of failures



→ **Cat!**

Thank you for your attention!



www.cern.ch