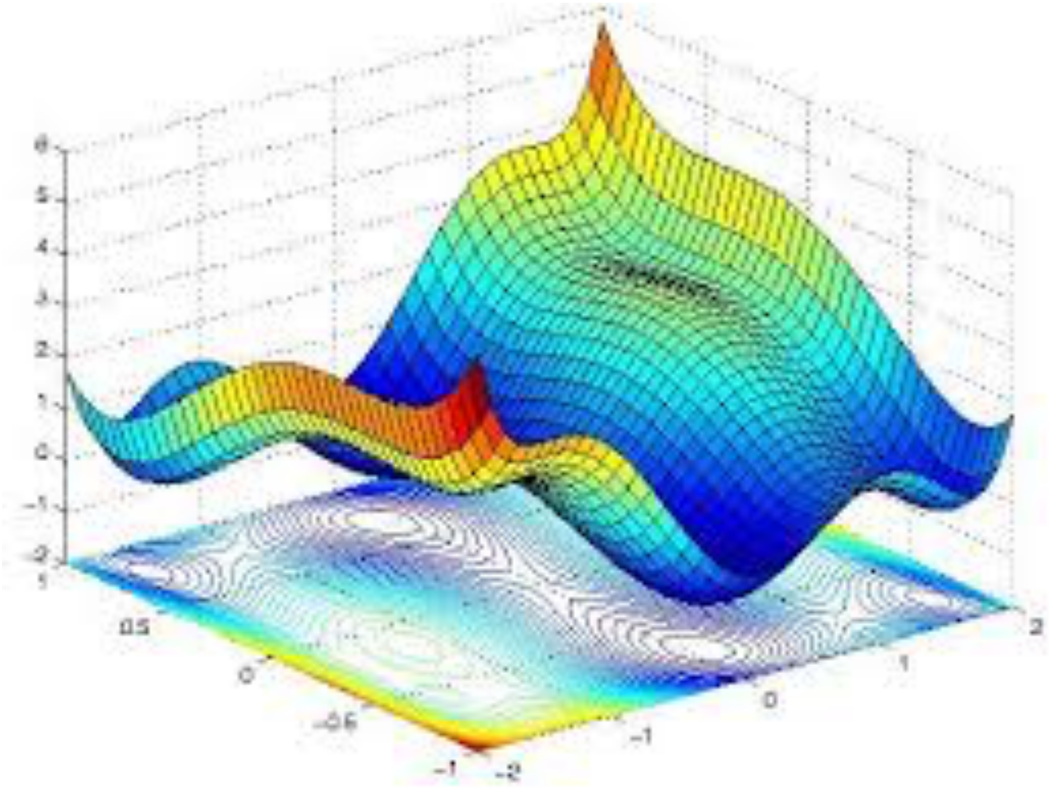
Machine learning opportunities

Maurizio Pierini

What is Machine Learning

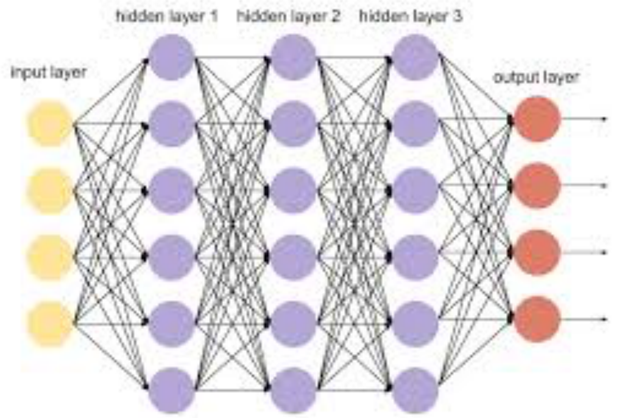
- *Machine learning is a technique by which an algorithm learns by example to accomplish a task*
  - *as opposed of being programmed to do so*
- *Deep Learning is the cutting-edge ML technology*
  - *based on "old-school" neural networks + augmented computational capabilities (e.g. GPUs)*
  - *the breakthrough is fast differentiability (back-propagation) allowing fast optimization*
  - *Learning non-linear functions, can be a fast shortcut to replace heavy processing tasks*

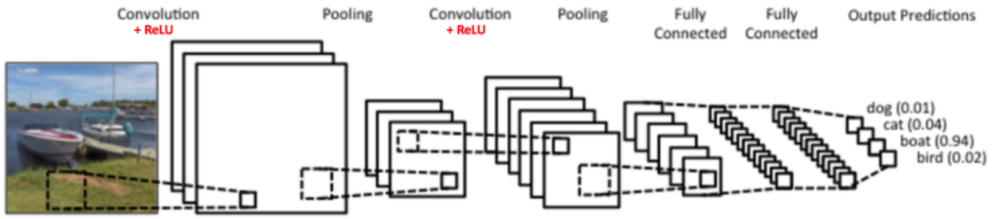


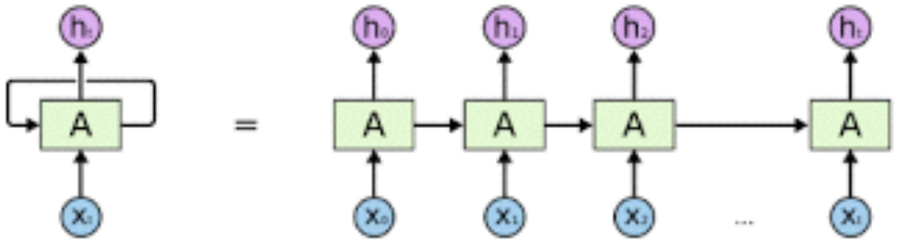
Training a Machine Learning algorithm consists in minimizing a complicated multi-dimensional function

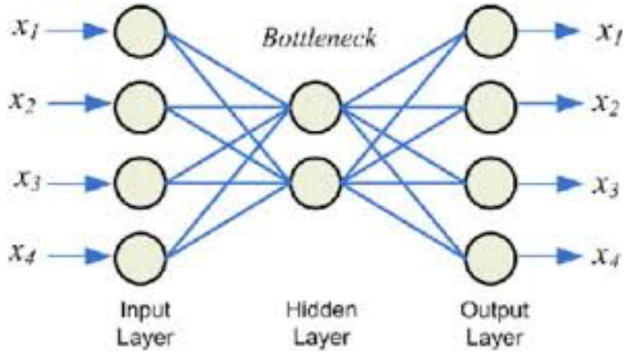
Deep Learning

- *New architectures proposed beyond classic fully-connected layers*
  - *Convolutional neural networks for image processing*
  - *Recursive neural networks for text processing*
  - *Generative adversarial networks for data generation*
  - *Autoencoders for anomaly detection (and data generation etc)*



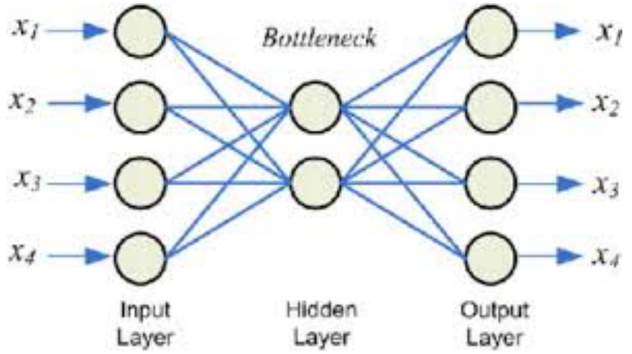






Directions for EP R&D

- *New architectures proposed beyond classic fully-connected layers*
  - *Convolutional neural networks for image processing*
  - *Recursive neural networks for text processing*
  - *Generative adversarial networks for data generation*
  - *Autoencoders for anomaly detection (and data generation etc)*

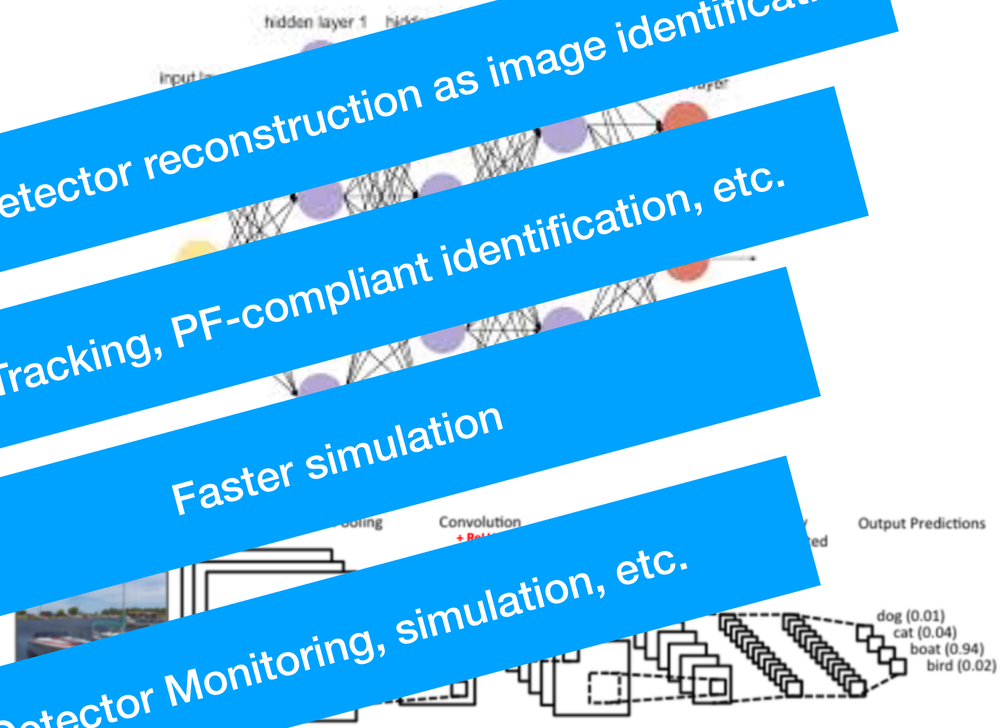


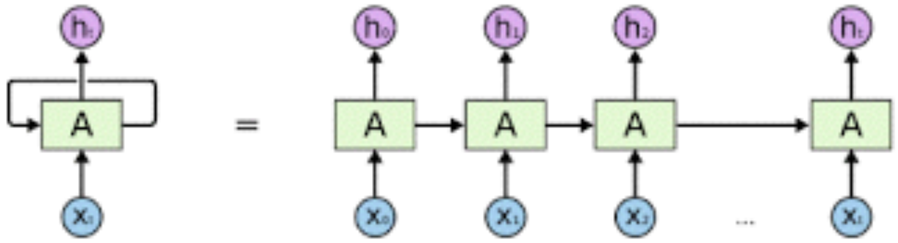
Detector reconstruction as image identification

Tracking, PF-compliant identification, etc.

Faster simulation

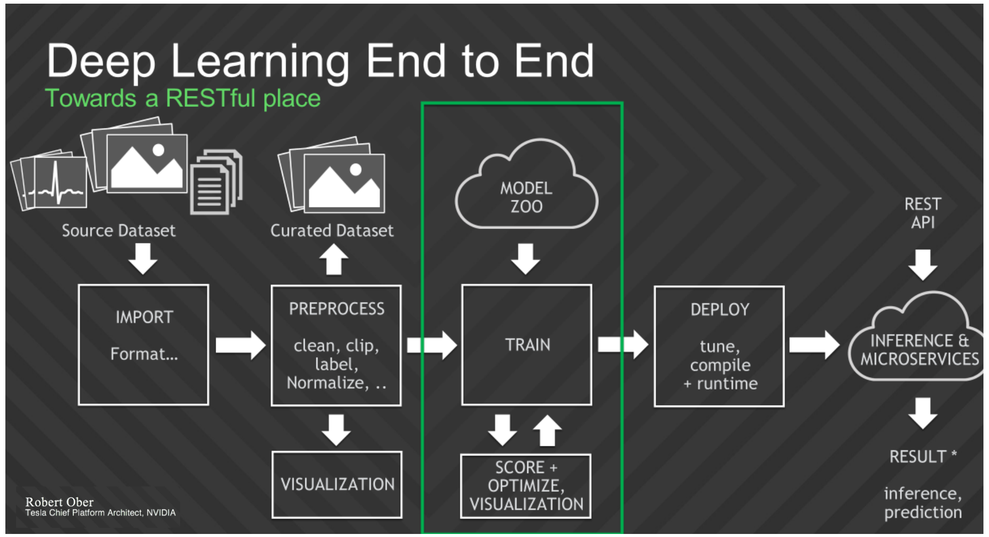
Detector Monitoring, simulation, etc.





Acquiring critical expertise

- *ML has been used for decades (mainly BDTs @LHC)*
- *Experiments transitioning to Deep Learning since a few years*
  - *Mainly through the push of young PhD students who studied this @school*
  - *Several initiatives @CERN: iML group, Data Science seminars, Conferences and Schools, Data Challenges, Marie Curie networks, one ERC grant*
- *So far, focus is more on the CERN users as a community of data scientists developing models. For a sustainable investment in this direction*



- ***Gain expertise within the Department***
- ***Go beyond the model training: data pipeline, model development, inference (i.e., using the models in what we do)***

Not just Model development

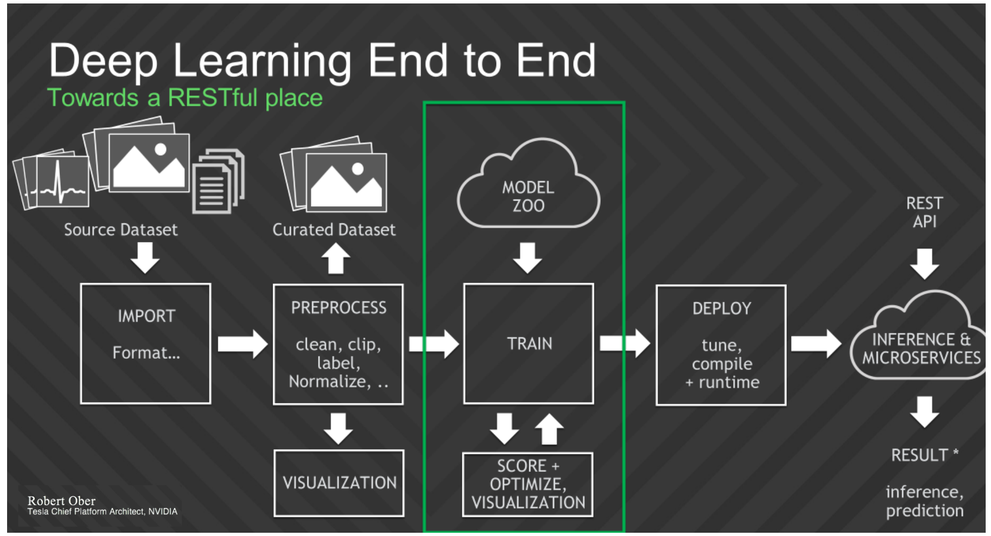
- *Training infrastructure*
  - *Parallel problem, runs efficiently on parallel architectures (e.g., GPUs, FPGAs, and TPUs) with out-of-shelf software solutions (python data science libraries)*
  - *Many groups (ATLAS, CMS, LHCb, ALICE) have in-house resources and recipes where proof-of-principle studies are running.*
  - *A user-oriented structured and general solution will be needed*
- *Porting Inference to real-life activity*
  - *Can run efficiently on modern parallel architectures with less resources request*
  - *But has to be integrated into our frameworks. Customisation needed at this stage*
    - *A model out-of-the-box from TensorFlow takes +100MB of memory. Smarted ways exist, but need to be developed and adapted to our needs*
  - ***Need to develop DL-compliant next-generation software (and possibly DL-compliant next-generation detectors, trigger farms, etc)***

Backup

Inference

Motivation

- Current issues:
  - » Models delivered by scientists way to large (easily >100MB)
  - » Training engines (often used in physics analysis) do not fit production environment (yes, people even invoke Keras from C++)
  - » Default inference engines (Tensorflow C++) do not integrate well with our processing frameworks
    - Manage Threads by themselves for instance
  - » Many new "COTS" optimized inference engines are built to run on cheap or even custom HW
- R&D:
  - » Tuning & compilation of Models
  - » Integration inference engines with processing frameworks
  - » Deployment of heterogeneous solutions in our production environment



Fast Inference

- *Neural network can model non linear functions*
  - *the more complex is the network, the more functions it can approximate*
- *Neural network are faster to evaluate (inference) than typical reco algorithm.*
  - *This is the speed up we need*
- *Neural Networks (unlike other kind of ML algorithms) are very good with raw (non-preprocessed) data (the recorded hits in the event)*

$$(pT, \eta, \phi, E)_{OFFLINE} = f\left((pT, \eta, \phi, E)_{ONLINE}\right)$$

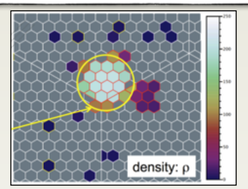
  - *could use them directly on the detector inputs*

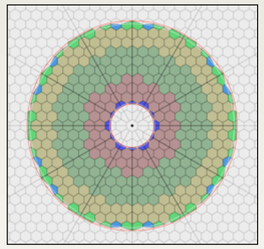
$$(pT, \eta, \phi, E)_{OFFLINE} = g\left(\text{Event hits}\right)$$

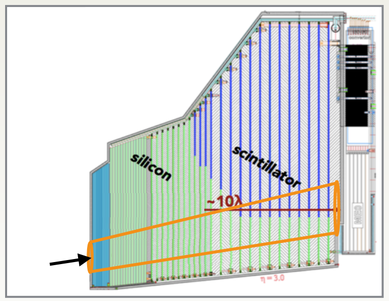
One would have to learn f and g to evaluate them at trigger. Online processing is replaced by offline training

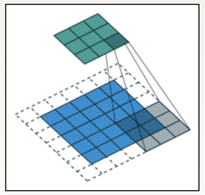
ML & (new) detectors

Map Detector Structure to Neural Network





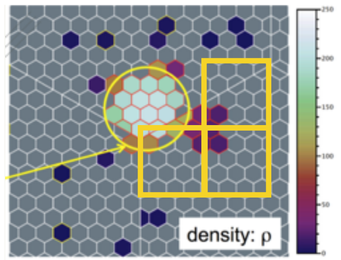


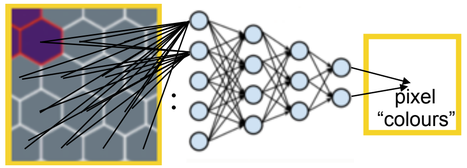


- Sensors hexagonal
- Sensor size/area changes with z,x,y
  - Physics based

- Uniform pixel size in all dimensions

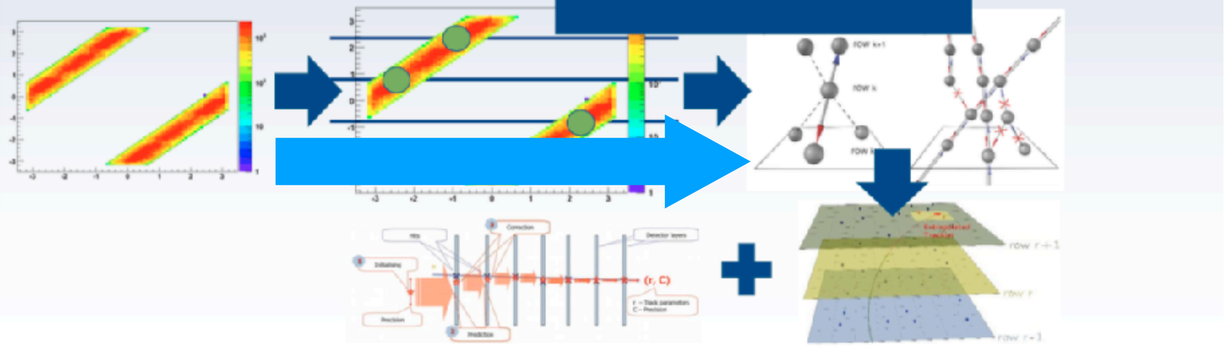
➡Correct representation of the geometry is an issue for **any** non-uniform non-squared sensor design





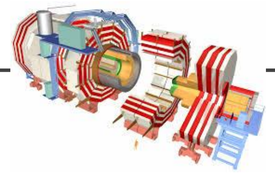
- The normal reconstruction chain is:
  - Clustering, tracking, vertexing, some higher level reco
  - I.e., in the TPC we start with a three-dimensional cha… heatmap is 2D).
  - The detector has some layers (2d planes in the 3d sp…
  - We forget about the original charge, and try to conne… GPUs are good at this).
  - Finally, we fit track parameters and extend (track follo… filter).

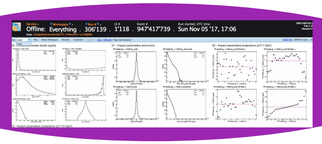
Why don’t we use the charge directly?



Data Quality

- **Real time feedback** on data quality to the shift crew





Online Histograms

ML model

Alarm ?

- Semi-supervised deep learning models are trained on past histograms to encapsulate the nature of “good data”
- Histograms from live monitoring are evaluated to identify problems with the hardware or the data taking conditions

- **Offline certification** of reconstructed data

RAW data

Hits and Particle candidates

ML model

Per lumi section flag

- Semi-supervised deep learning models trained on initial runs to encapsulate the nature of “good data”
- Exploit the entire statistical power and full event reconstruction to single out time intervals affected by failures or bad performanve
- Replacing labor intensive run level selection

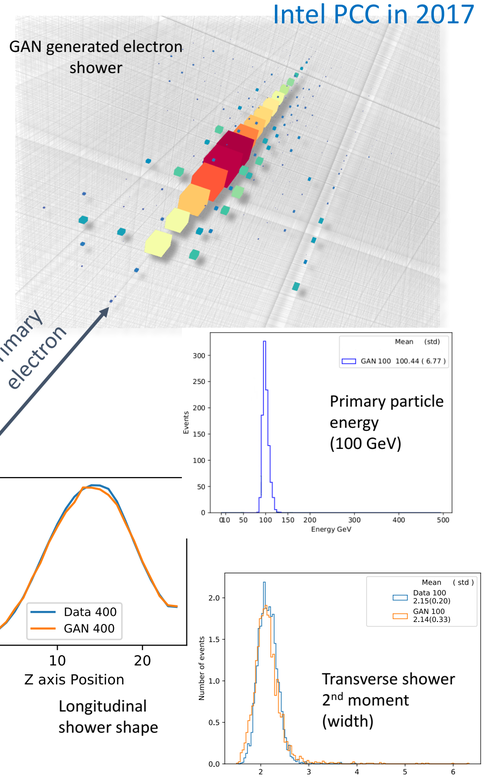
Simulation

Current status

Intel PCC in 2017

First prototype: calorimeter simulation as 3D image generation

- CLIC electromagnetic calorimeter (high granularity)
- 3D Convolutional Generative Adversarial Networks
- Realistic generation of samples
- Detailed physics validation
  - Comparison to full sim
- Relatively simple prototype but very promising approach



Many activities started in experiments along the same research line!

CERN openlab intel

YES ML, BUT HOW?

- Machine learning applied to FASTSIM looks very promising

- What if we go one level beyond and we replace computationally expensive physics models with ML blocks
  - Able to learn complex cross-sections shapes (total, differential)?
  - Able to directly generate the final-state?

  → From "physics-agnostic" to **"physics-aware"** neural networks

Training Physics-aware supervised neural networks[1][2]
- Embed physical-laws underlying the process
- To be used to infer physical quantities (momenta, directions, energies..)
- Both for continuous and discrete processes

[1] "QCD-Aware Recursive Neural Networks for Jet Physics", Kyle Cranmer et Al, https://arxiv.org/abs/1702.00748 - Feb 2017

[2] Physics Informed Deep Learning: Data-driven Solutions of Nonlinear Partial Differential Equations, Maziar Raissi et Al, https://arxiv.org/abs/1711.10561 - Nov 2017

Learning with uncertainties

- *New techniques are developed to deal with the uncertainties of the network training*
- *Notable examples by people from HEP (https://arxiv.org/pdf/1611.01046.pdf)*

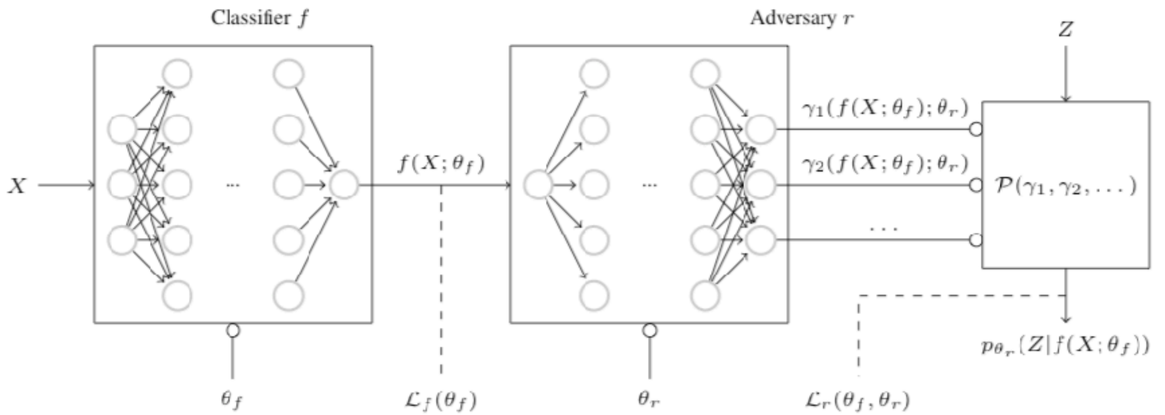


Figure 1: Architecture for the adversarial training of a binary classifier f against a nuisance parameters Z. The adversary r models the distribution $p(z|f(X;\theta_f)=s)$ of the nuisance parameters as observed only through the output $f(X;\theta_f)$ of the classifier. By maximizing the antagonistic objective $\mathcal{L}_r(\theta_f,\theta_r)$, the classifier f forces $p(z|f(X;\theta_f)=s)$ towards the prior $p(z)$, which happens when $f(X;\theta_f)$ is independent of the nuisance parameter Z and therefore pivotal.

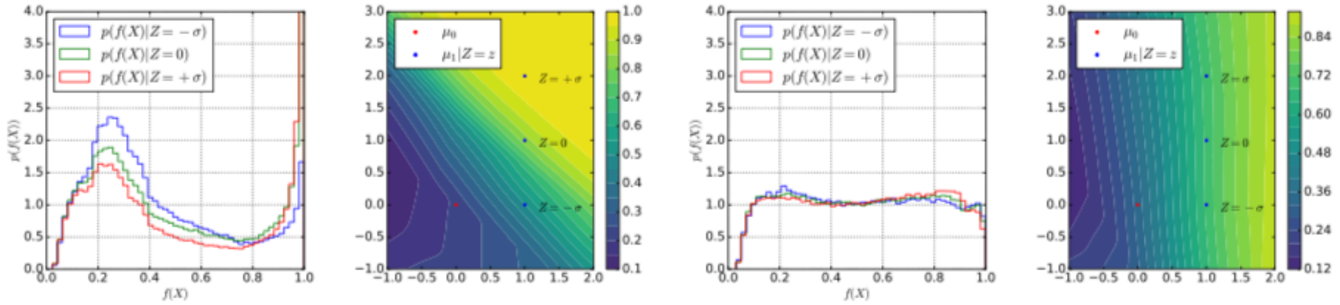


Figure 2: Toy example. (Left) Conditional probability densities of the decision scores at $Z=-\sigma,0,\sigma$ without adversarial training. The resulting densities are dependent on the continuous parameter Z, indicating that f is not pivotal. (Middle left) The associated decision surface, highlighting the fact that samples are easier to classify for values of Z above σ, hence explaining the dependency. (Middle right) Conditional probability densities of the decision scores at $Z=-\sigma,0,\sigma$ when f is built with adversarial training. The resulting densities are now almost identical to each other, indicating only a small dependency on Z. (Right) The associated decision surface, illustrating how adversarial training bends the decision function vertically to erase the dependency on Z.