LABORATÓRIO DE INSTRUMENTAÇÃO
E FÍSICA EXPERIMENTAL DE PARTÍCULAS
partículas e tecnologia

anos 1985 · 2025 Portugal CERN

[MACHINE LEARNING *in Particle Physics*]

Rute Pedro | 18th June | Lectures and Tutorials LIP Internship 2025

CERN/FIS-PAR/0010/2021

Fundação para a Ciência e a Tecnologia

Lisb@20[20]

COMPETE 2020
PROGRAMA OPERACIONAL COMPETITIVIDADE E INTERNACIONALIZAÇÃO

PORTUGAL 2020

UNIÃO EUROPEIA
Fundo Europeu de Desenvolvimento Regional

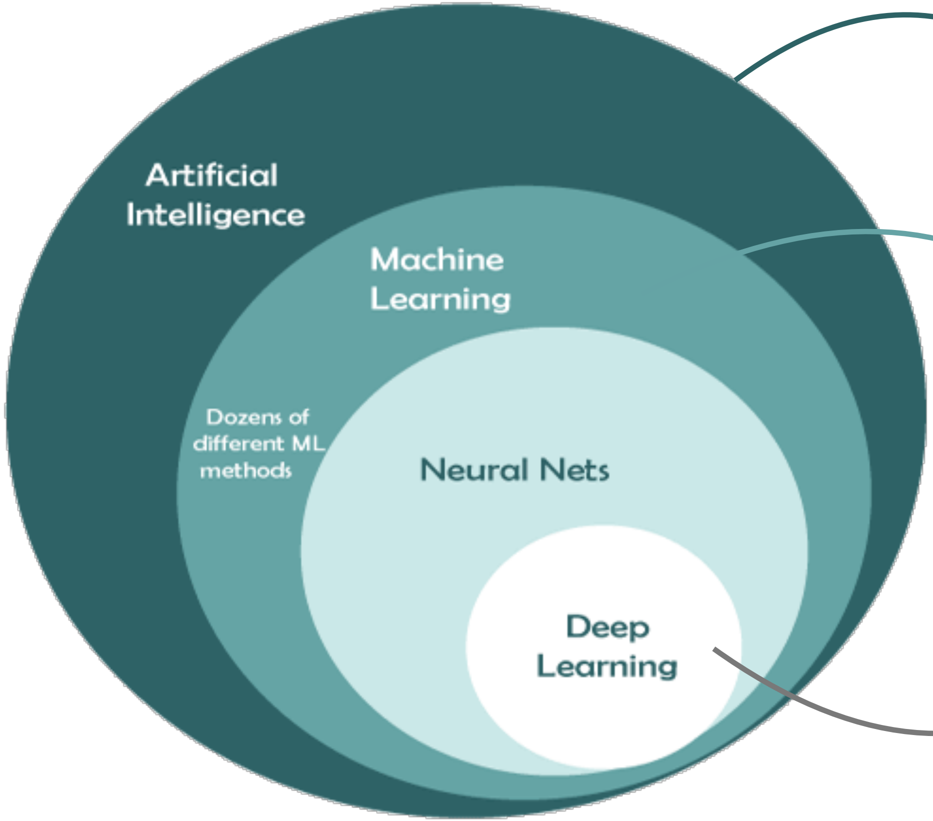
ATLAS EXPERIMENT

Outline

Machine Learning: key concepts

ML applications to Particle Physics

What is Machine Learning?



Ability of a machine to imitate intelligent human behaviour

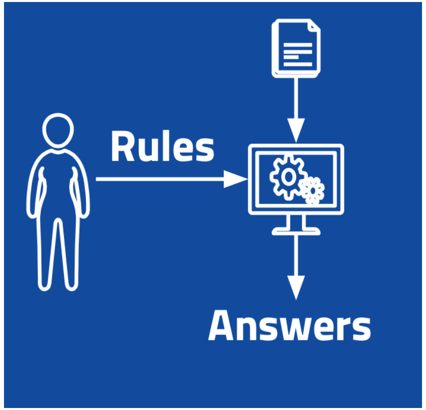
Application of AI that allows a system to learn automatically to perform tasks

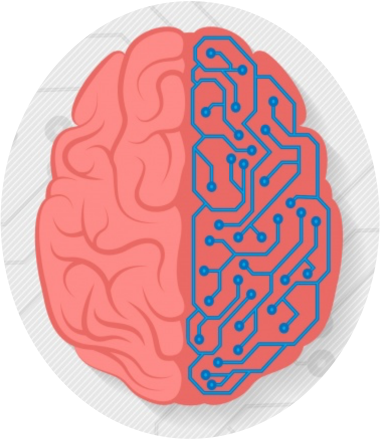
Machine Learning based on Artificial Neural Networks

What is Machine Learning?

Traditional Computation

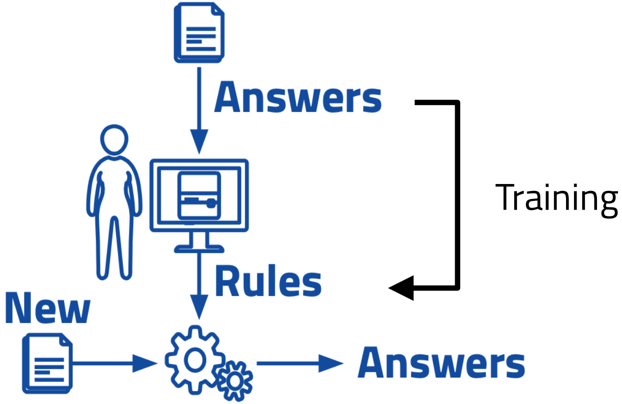
The task is programmed by the user as a pre-defined set of rules/algorithms to apply to data





Machine Learning (ML)

The program learns from data what are the necessary rules to execute a task/objective defined by the user: Training



From M. C. Romão [LIP seminar]

What is Machine Learning?

Example: **Linear Regression**!

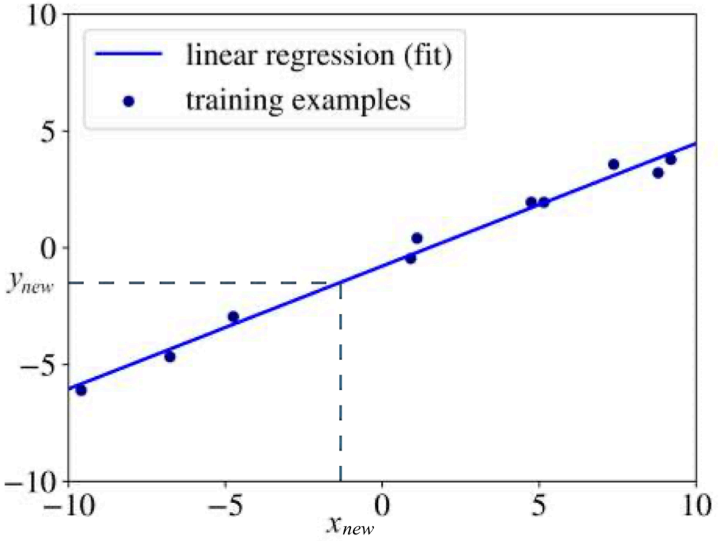
Process of solving a practical problem by:

1. **Gathering a dataset** $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$

$\mathbf{x}_i$ - "feature vector": properties of the example i

x^j - individual feature

y_i - "label": element of a class set or real number

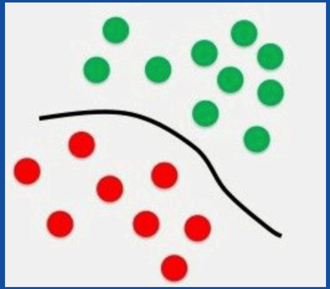


2. **Algorithmically building a statistical model based on that dataset**

ML tasks

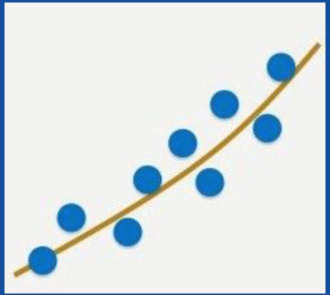
Classification

Discrete prediction



Regression

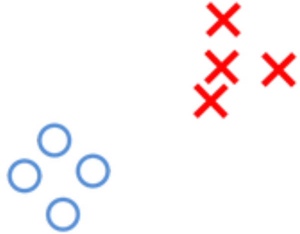
Real-value prediction



Learning types

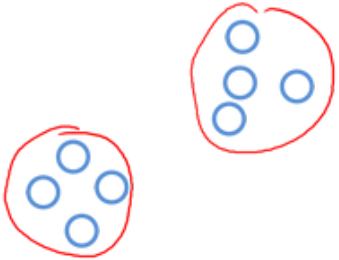
Supervised

(E.g. Simulation in Particle Physics)

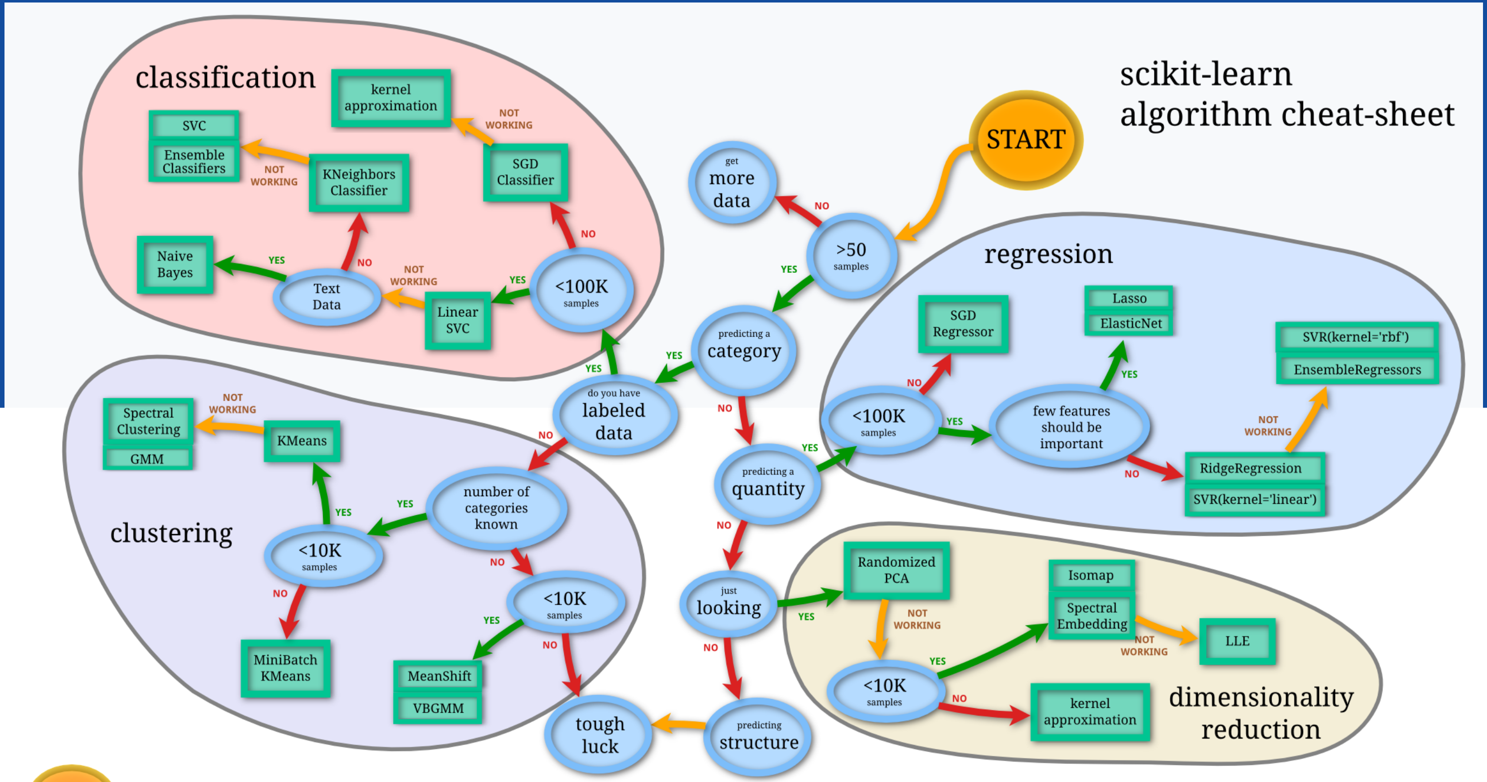


Unsupervised

(E.g. clustering)



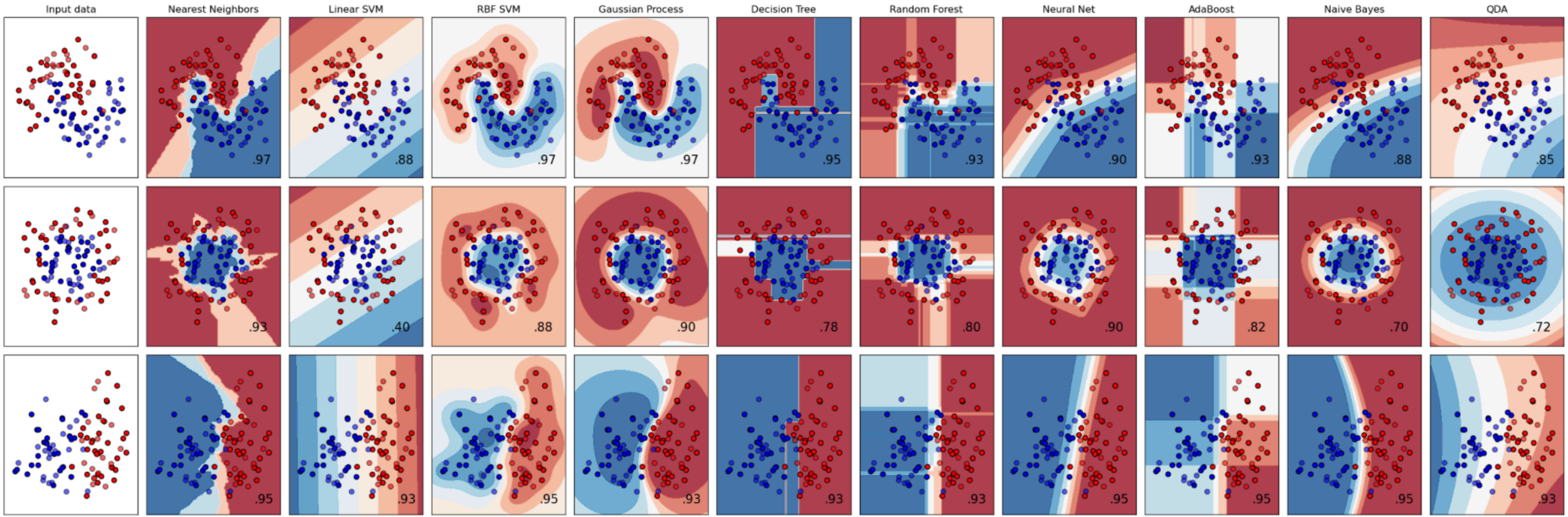
... an entire ecosystem



Scikit-Learn: excellent ML library to start with, Python-based

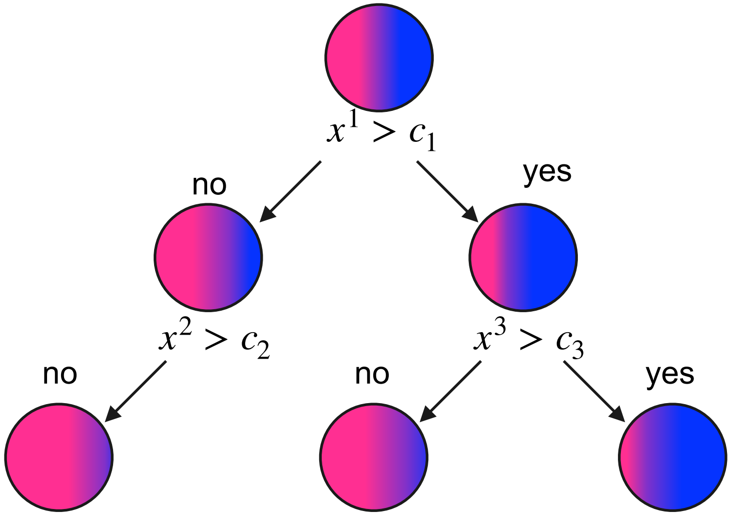
Besides algorithms, it also contains data

Miscellanea of Algorithms



https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

Shallow Learning Decision Tree



$\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ samples of labeled data

Partitions the data to increase sample purity

Finds optimal criteria $x^i > c_i$ to separate data categories

Category prediction based on the label of the majority samples of the end leaf

User-defined hyper-parameters (tree depth, …)

Very popular algorithm

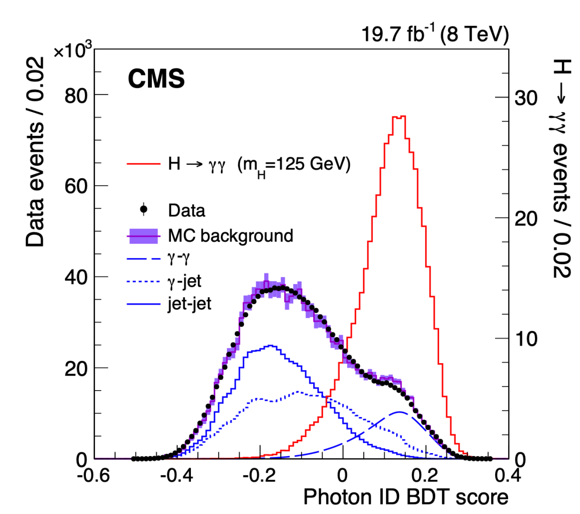
"Non-parametric" algorithm, i.e. no $(\mathbf{w}*, b*)$

Observation of $H \to \gamma\gamma$ in CMS

1407.0558

Boosted Decision Trees used in many aspects of the analysis

- Selection of collision vertex
- Photon identification
- Photon energy corrected with BDT regression
- Several BDT to extract signal in different categories
- ...



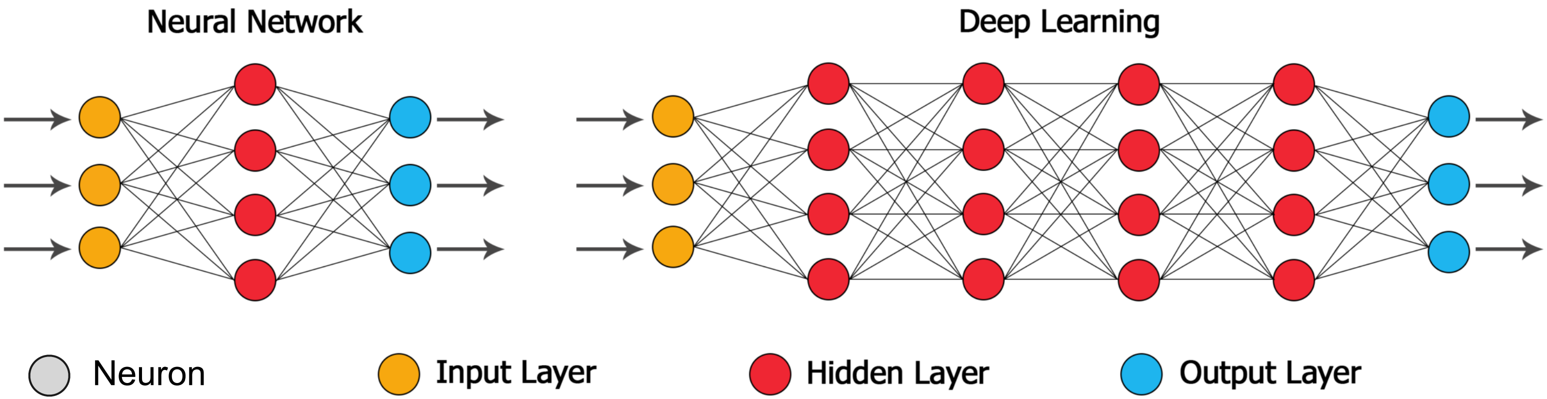
PHOTON IDENTIFICATION

- BDT discriminates photons from fakes (π^0):
  - Shower shape and isolation variables
  - Photon p_T, η

Signal observed with 5.2σ significance

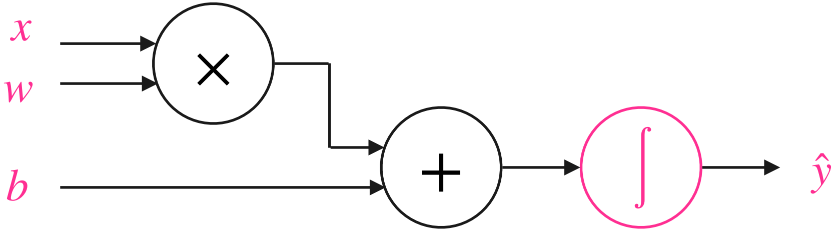
ML impact on signal sensitivity equivalent of 50% more data

Deep Learning



- Neural networks with many hidden layers, each with a given number of artificial neurons
- Capable of highly non-linear representations of the data
- In principle, can model any function
- Architecture -> hyper-parameters: number of layers, number of neurons/layer, ...

Artificial Neuron

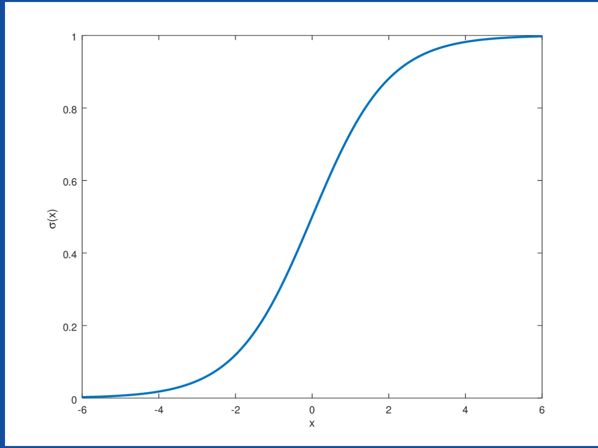


- x is the input feature
- y is the target feature (or “label”)
- w, b are the model trainable parameters
- $\hat{y} \equiv f(x, w, b)$ is the output (model prediction)

$\int$ Activation function

- e.g. linear for regression
- e.g. sigmoid for classification

$$f(x) = \frac{1}{1 + e^{-x}} \rightarrow \hat{y}$$



Artificial Neural Network Training

Loss function L: measure of how good is $\hat{y}$ in predicting y (true value)

- e.g. Mean squared error: $L = \frac{1}{N} \sum_{i}^{N} (y_i - \hat{y}_i)^2$
- e.g. Binary cross-entropy: $L = \frac{1}{N} \sum_{i}^{N} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i)$

Training objective: find w, b that minimise the Loss function

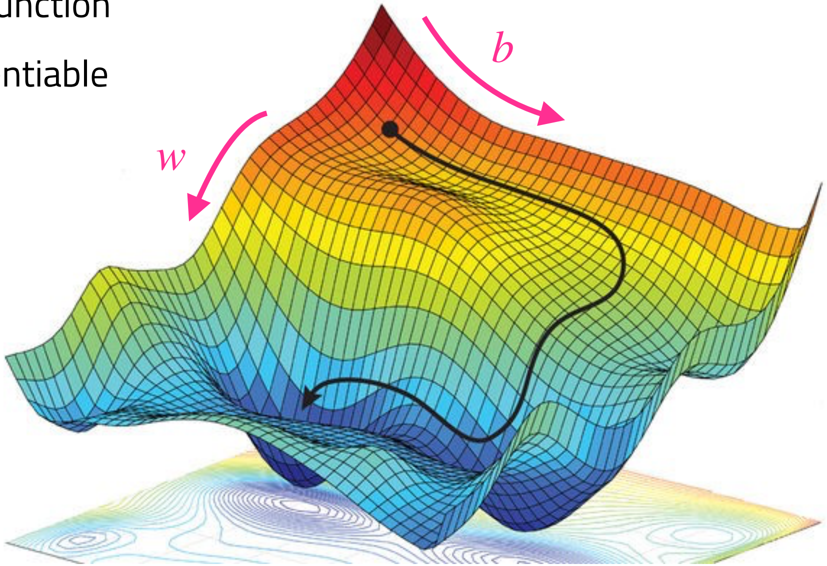
Gradient Descent

Iterative optimisation algorithm to find the minimum of a function

Most frequently used when optimisation criterion is differentiable

Consider $L = f(\mathbf{x}, w, b)$

Gradient descent consists of rolling down the surface



Optimisation criterion surface

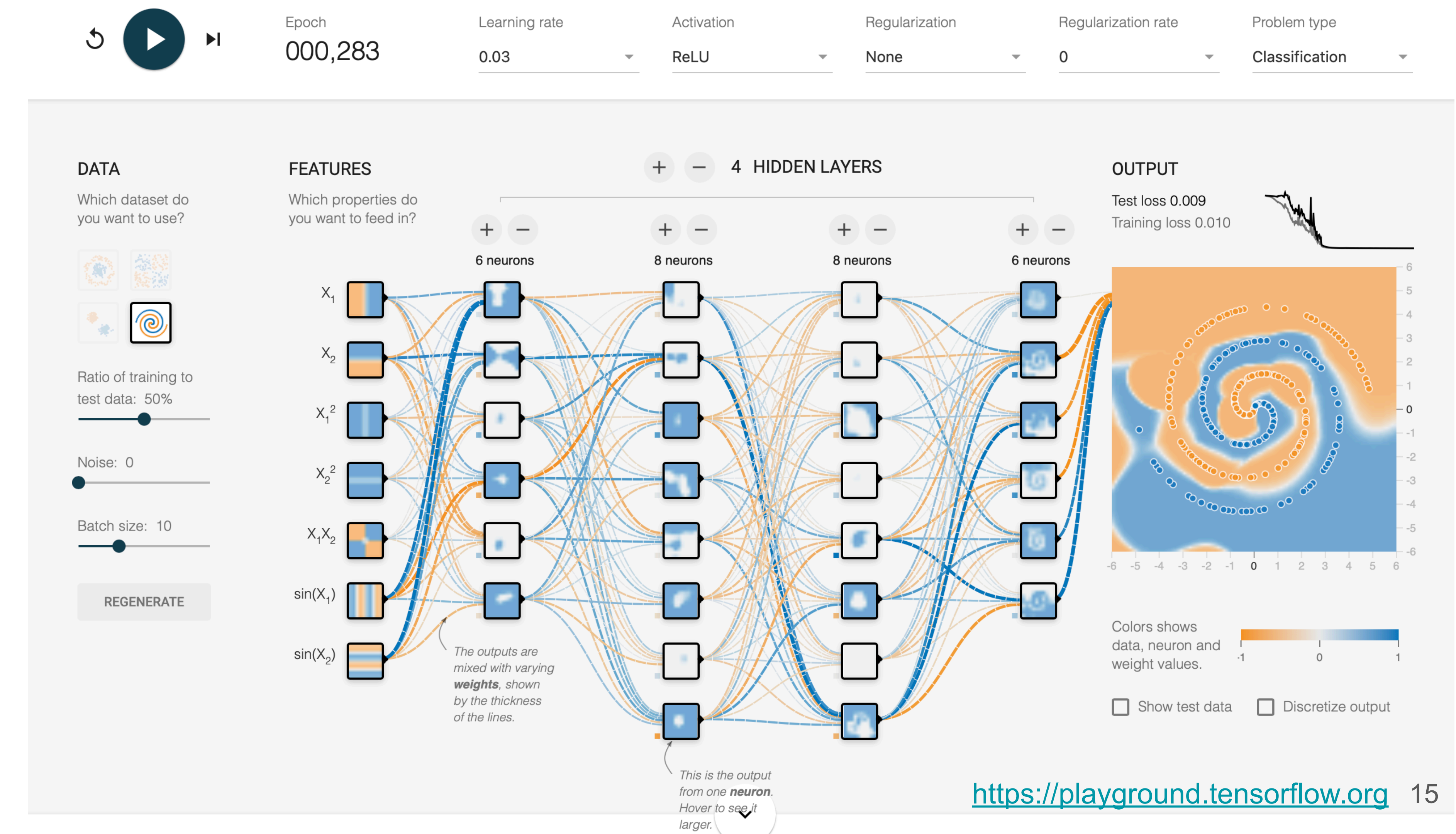
Compute and **Back-propagate** the gradient (iteratively)

Compute $\frac{\partial L}{\partial w}$ and update $w \leftarrow w - \alpha \frac{\partial L}{\partial w}$

Compute $\frac{\partial L}{\partial b}$ and update $b \leftarrow b - \alpha \frac{\partial L}{\partial b}$

α is an hyper-parameter that adjusts the **learning rate**

Each iteration is called a training "**epoch**"



https://playground.tensorflow.org

Practicable Deep Neural Networks

Many layers + many units

- **Vanishing gradient**: new activation functions made training possible (ReLU) (~2010)
- Advances in hardware: **GPU** increased speed of computation by 100 (~2010)
- APIs: **Keras** , **Tensorflow** (2015)

Deep learning

- Many parameters to estimate: $\{\vec{w}, \vec{b}\}$
- **Data** thirst

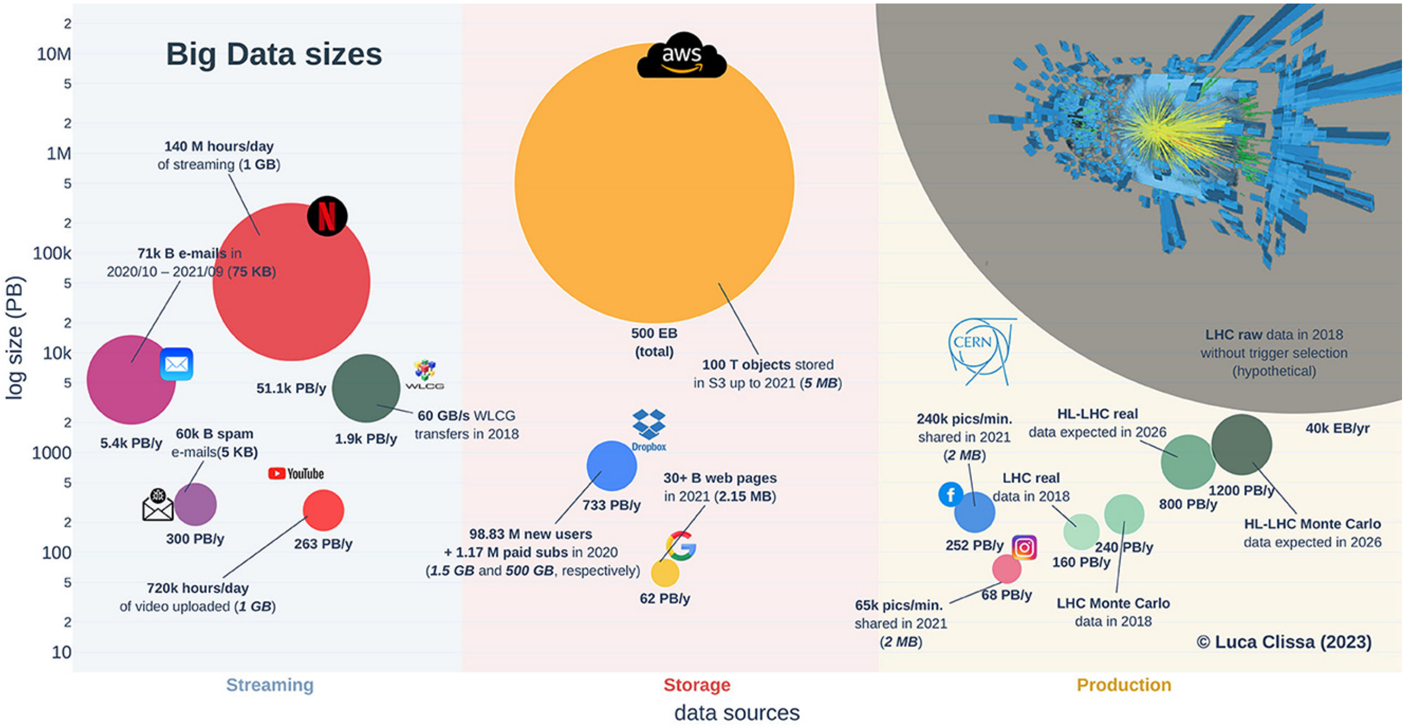
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten_10 (Flatten)         (None, 784)               0
_________________________________________________________________
dense_22 (Dense)             (None, 128)               100480
_________________________________________________________________
activation_19 (Activation)   (None, 128)               0
_________________________________________________________________
dense_23 (Dense)             (None, 128)               16512
_________________________________________________________________
activation_20 (Activation)   (None, 128)               0
_________________________________________________________________
dense_24 (Dense)             (None, 10)                1290
_________________________________________________________________
activation_21 (Activation)   (None, 10)                0
=================================================================
Total params: 118,282
Trainable params: 118,282
Non-trainable params: 0
_________________________________________________________________
```

Outline

Machine Learning:
key concepts

ML applications to
Particle Physics

Particle Physics as a Big Data playground for ML



E.g. LHC is an enormous source of data

- Number of collisions: 40 MHz, 1kHz recorded
- High data dimensionality: O(100 M) readout units

Lower rate Particle Physics experiments

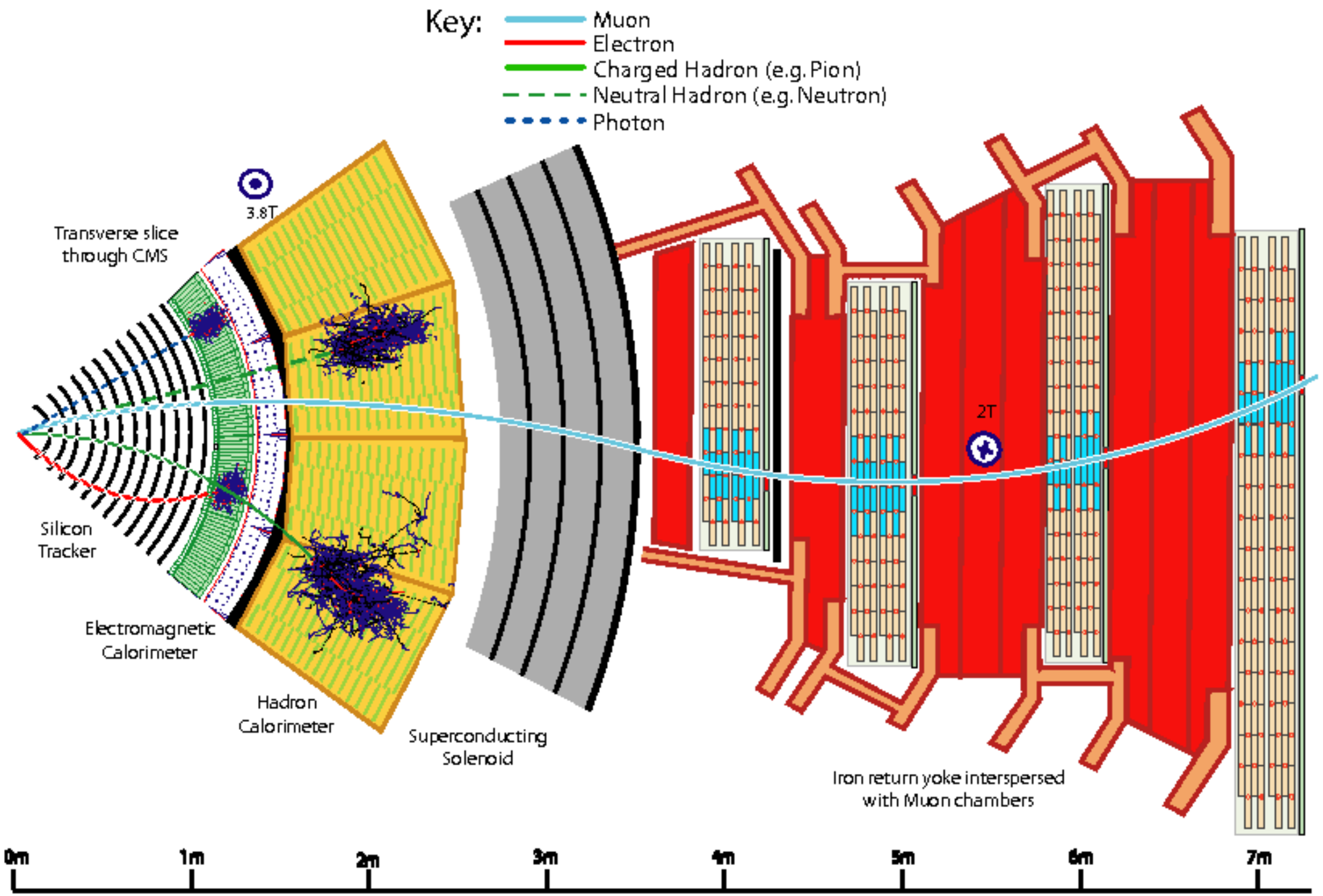
- Large simulation datasets to train ML
- Applied to real data

Anatomy of a HEP event

LHC example

- Identify collision vertices and particles:
    - Track-finding
    - Electron/jet/muon ID/reconstruction
- Measure energy, momenta, electric charge
- Jet flavour?
- Signal topology?

ML is key in many of these tasks



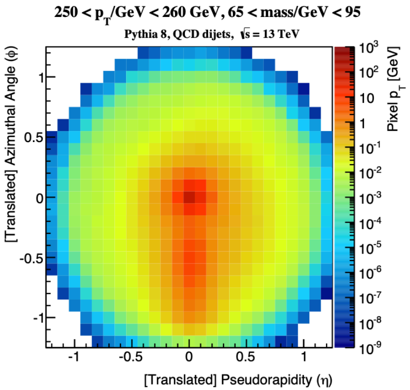
How to represent data?

… part of the definition of the ML algorithm

Tabular

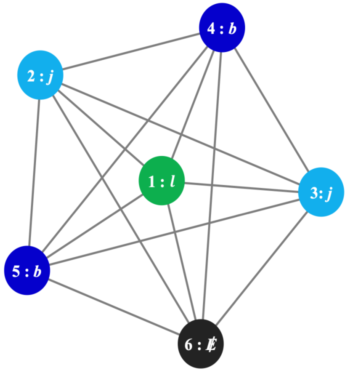
	Electron1_PT	FatJet1_PT	Jet1_PT	Muon1_PT
0	227.793961	253.598358	254.124435	0.000000
1	0.000000	225.937729	228.712021	39.127575
2	68.204712	0.000000	144.771240	0.000000
3	133.825851	229.350952	219.542404	0.000000
4	0.000000	0.000000	127.972099	0.000000
5	82.530861	259.897095	206.621994	0.000000
6	0.000000	0.000000	119.139641	0.000000
7	170.190216	0.000000	199.339508	0.000000
8	0.000000	276.407806	275.428223	219.815781
9	43.247391	240.832916	240.927399	0.000000

Image



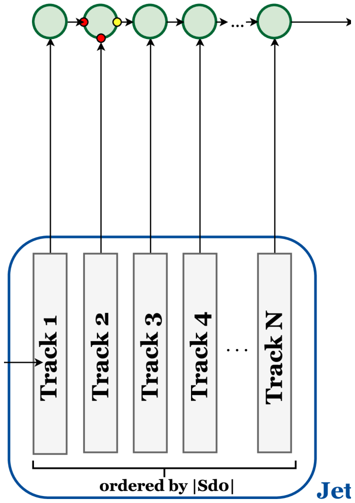
[arXiv:1511.05190]

Graph



[arXiv:1807.09088]

Sequences

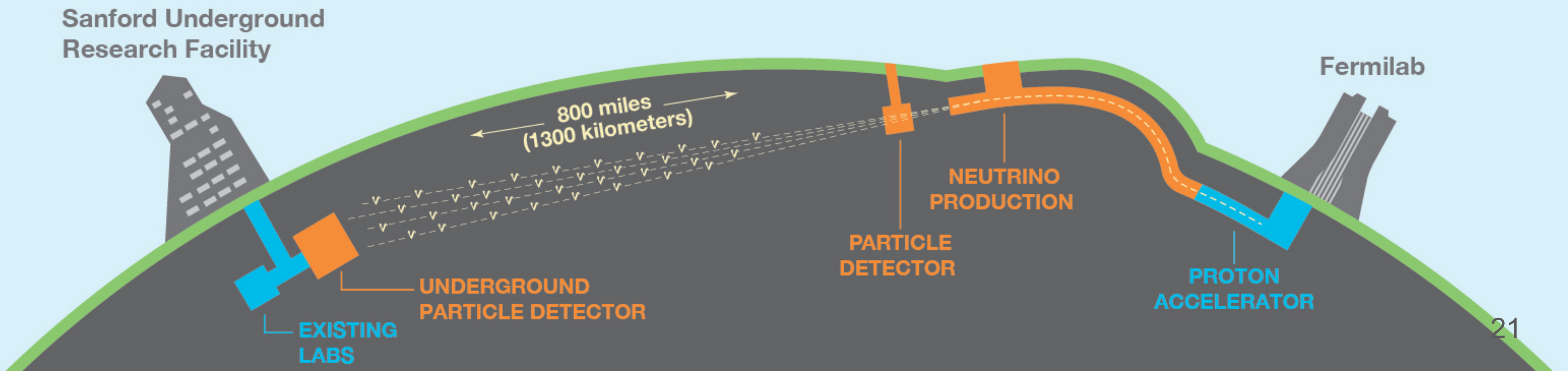


[ATL-PHYS-PUB-2017-003]

Convolutional NNs for Neutrino Flavour

DUNE being set to study neutrino oscillations

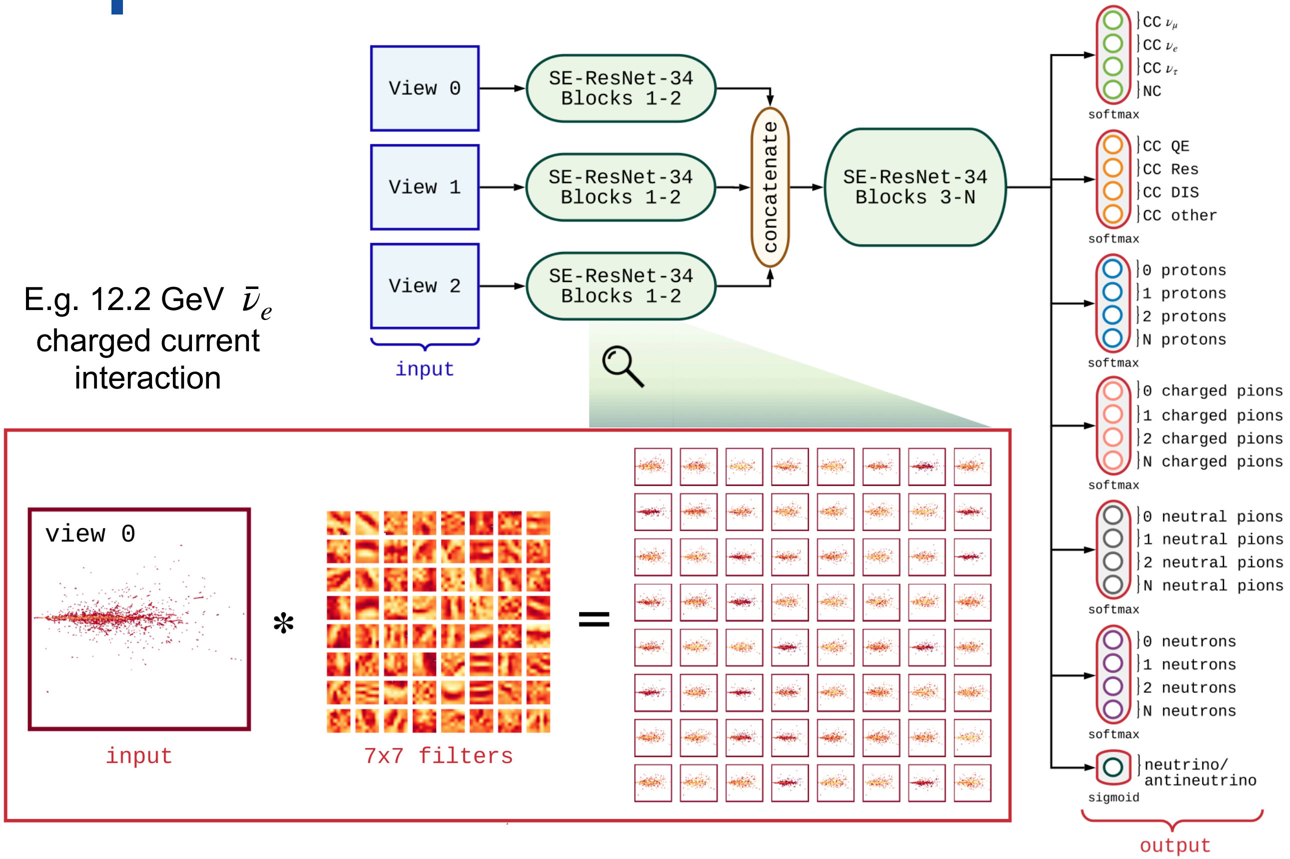
- Intense neutrino beam $\{\nu_\mu, \bar{\nu}_\mu\}$ - dominated
- Underground far detector with 70 kTon of liquid argon (DUNE)
- Determining the neutrino flavour is key to the experiment



Convolutional NNs for Neutrino Flavour

2006.15052

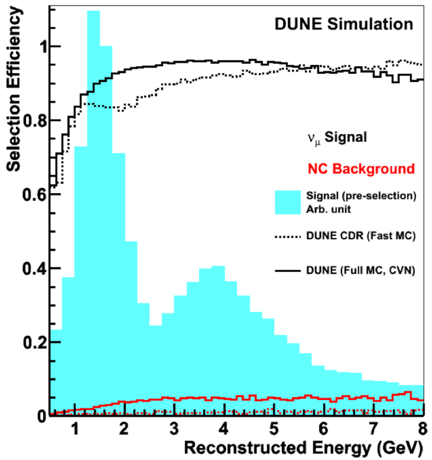
E.g. 12.2 GeV $\bar{\nu}_e$ charged current interaction



Multi-classification of **signal images**

- 35x35 pixel
- Signal time VS Detector wire
- 3 views/event

Convolutional filters look for discriminant patterns

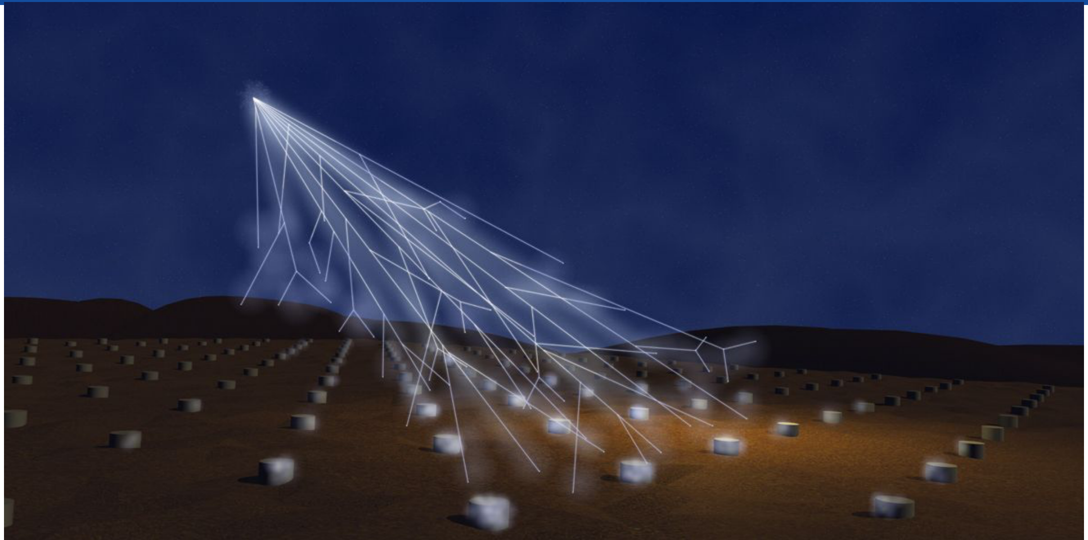


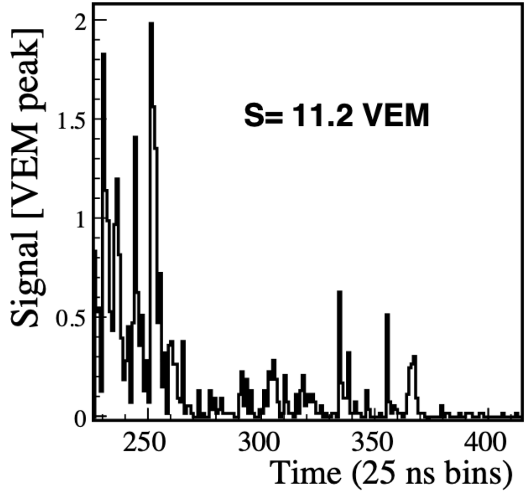
Cosmic ray composition with Genetic algorithms
Auger Observatory

PIERRE AUGER OBSERVATORY

- Array of water Cherenkov detectors covering 3000 km^2 to study cosmic rays ($E > 10^{18}$ eV)
- Infere properties/origin of primary particle from extensive air showers
- Determine **muonic component**
  - Validate shower simulation
  - Measure primary particle mass

Water tank signal has muonic and electromagnetic components





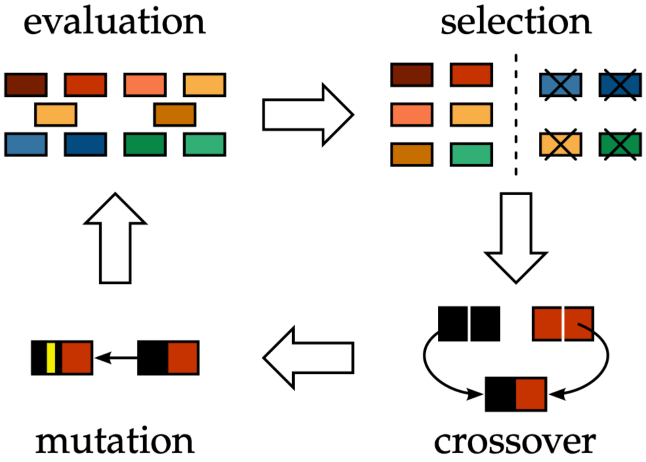
Cosmic ray composition with Genetic algorithms

Number of muons

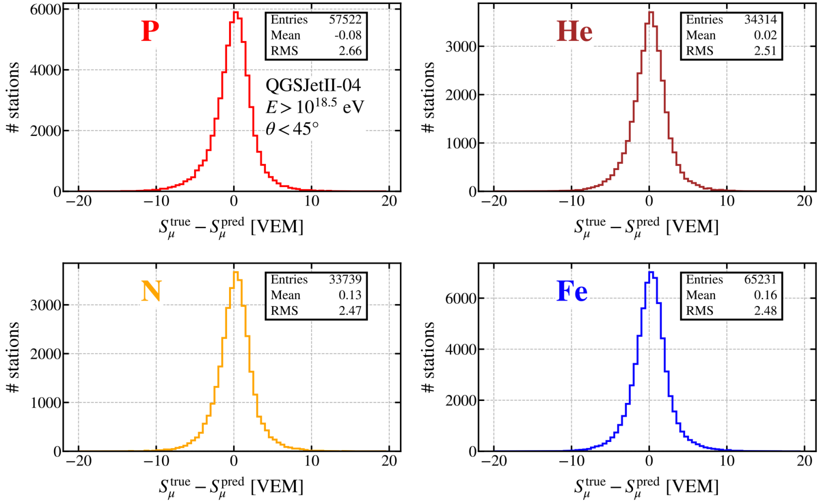
1807.09024

Regression DNN to find out the number of muons

- Hyper-parameters optimised with genetic algorithms
- Train a number of DNNs with different n° layers/n° neurons/activation functions
- DNN with better performance selected in binary tournaments, then crossed-over and mutated



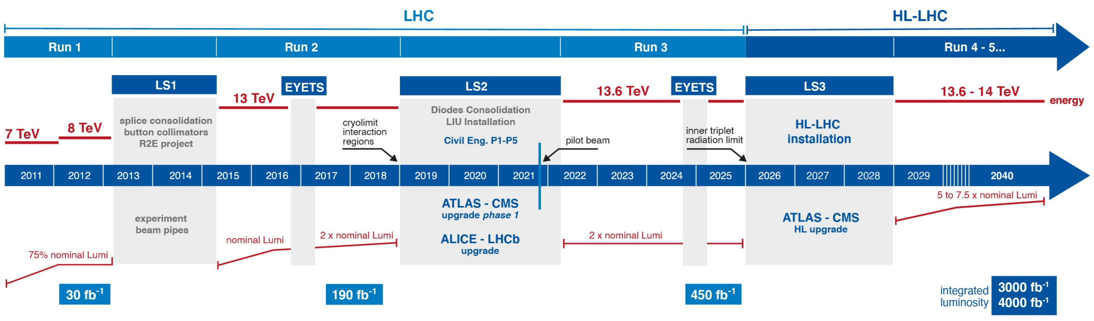
Accuracy better than 10%, good precision

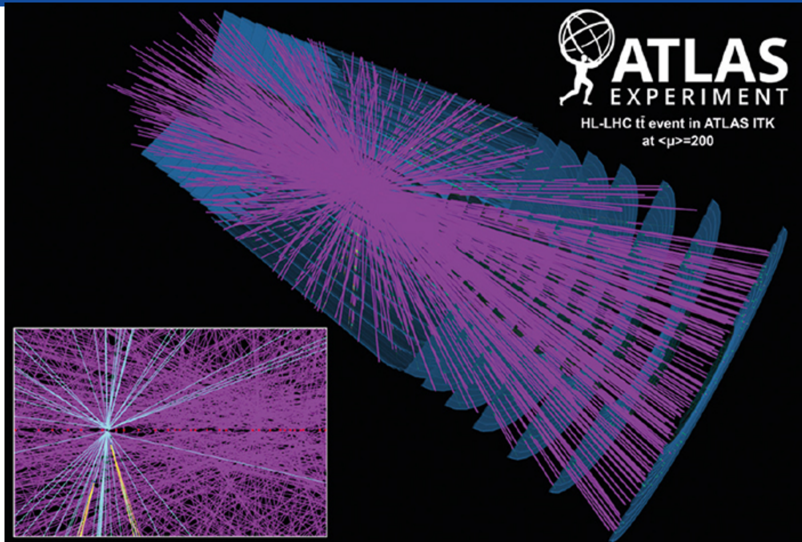


ML in the future of HEP
HL-LHC upgrade

Many challenges and opportunities where ML can be a handle

- High pile-up: collisions per bunch crossing $33 \to 140$
- Noisy environment: ambiguous track hits reconstruction, collision vertex finding, pile-up energy subtraction,...
- Big data phase: 3000 fb^{-1}, increased need for simulation



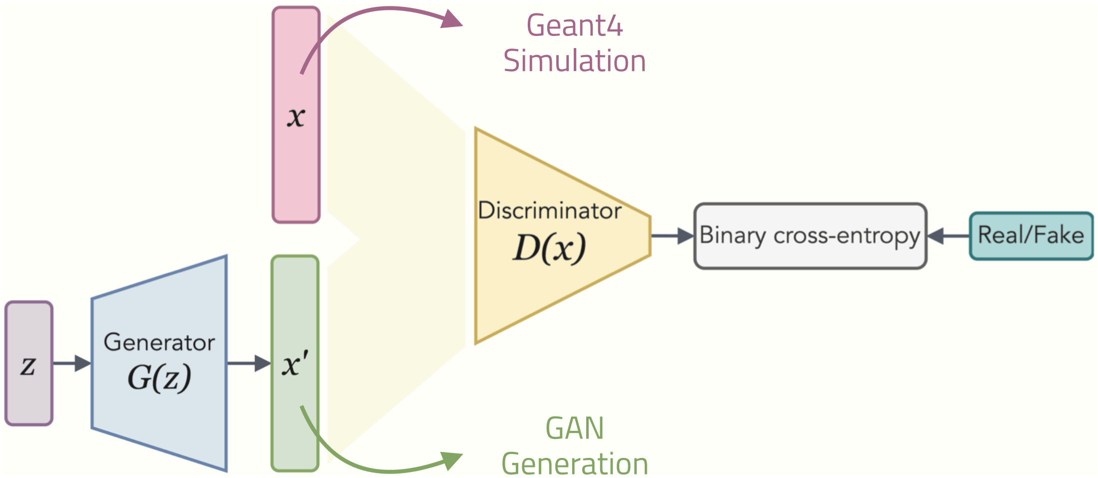


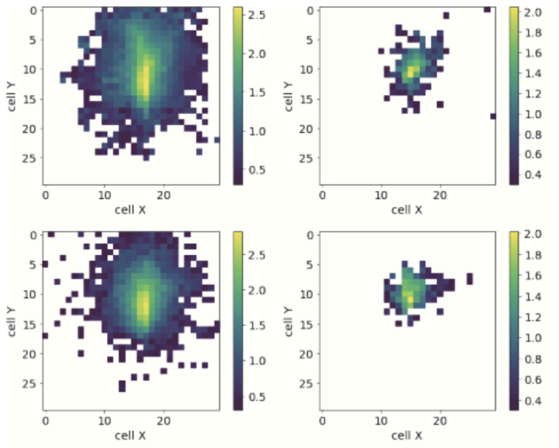
Calorimeter simulation

Generative AI application

Measurements rely on comparisons between data and simulation (~1000 M for a typical analysis)

- Calorimeter showering is the heaviest load (particle multiplicity and overlap)
- Generate synthetic showers given a particle and the calorimeter geometry
- Train the generator by comparing synthetic to Geant4 showers





(a) $E_0 = 63.7$ GeV

(b) $E_0 = 6.5$ GeV

Data Quality Monitoring

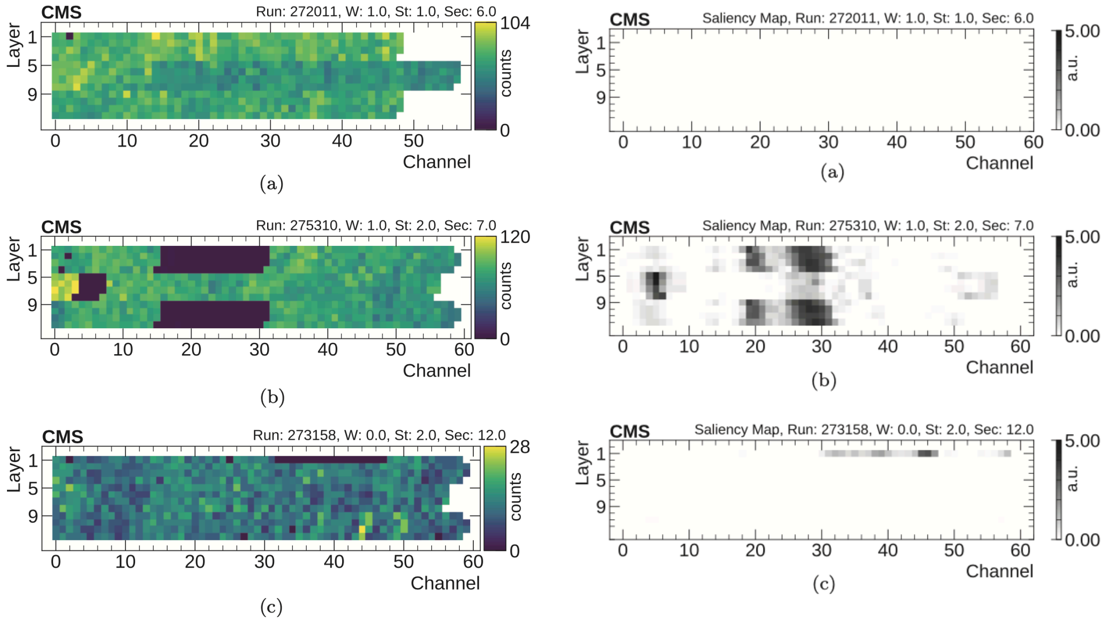
Automating defect detection

1808.00911

Pre-defined routines fail to recognise novel patterns of detector failure/defects and rely on heavy human supervision

Anomaly detection outperforms in identifying defects, regardless of previous knowledge

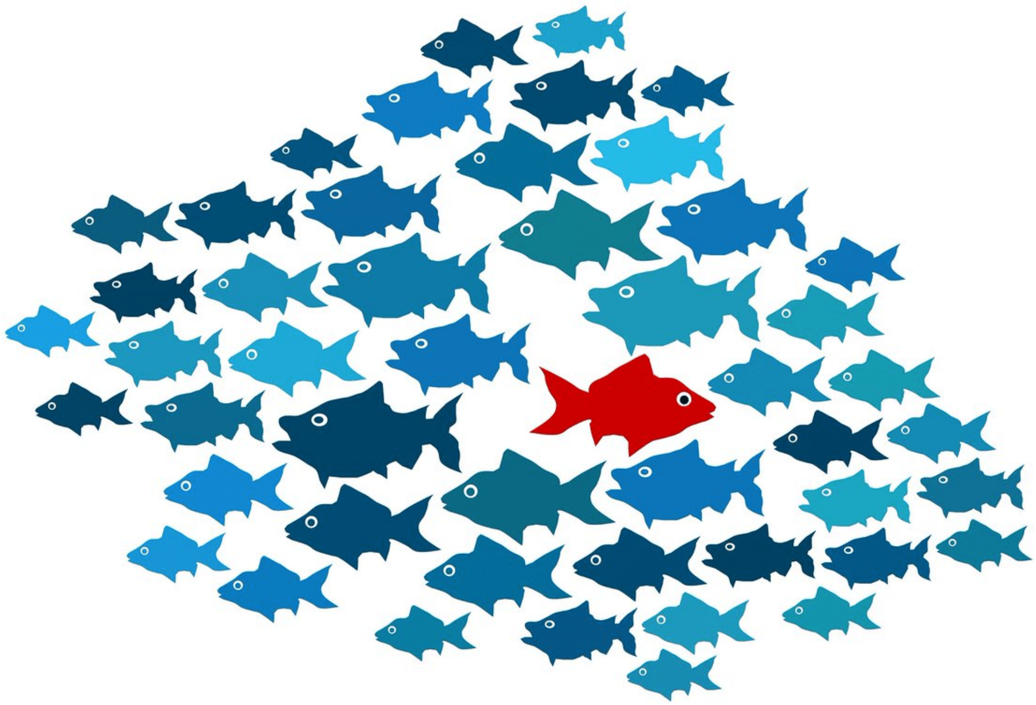
Eg. Based on the construction of saliency maps using Convolutional Neural Networks



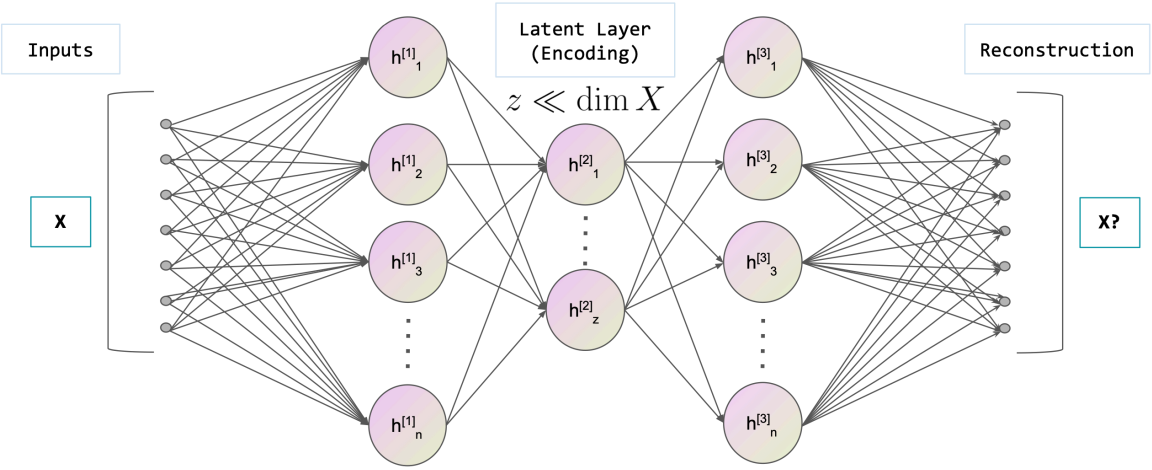
Anomaly detection in the search for New Physics

A primary LHC goal remains to conquer: no sign of New Physics so far!...

ML used in direct searches, classifiers trained to recognise specific signals

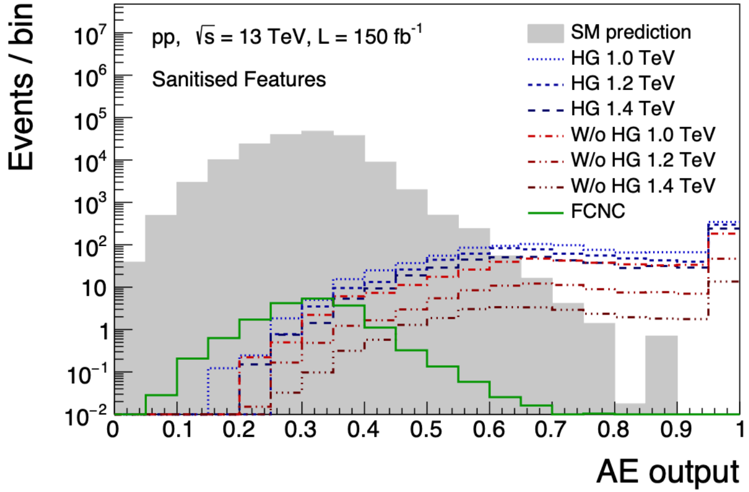


Auto-Encoder



- Training objective is to minimize input reconstruction loss
- More common events will be better reconstructed
- Reconstruction error is a measurement of anomaly/ *outlyingness*

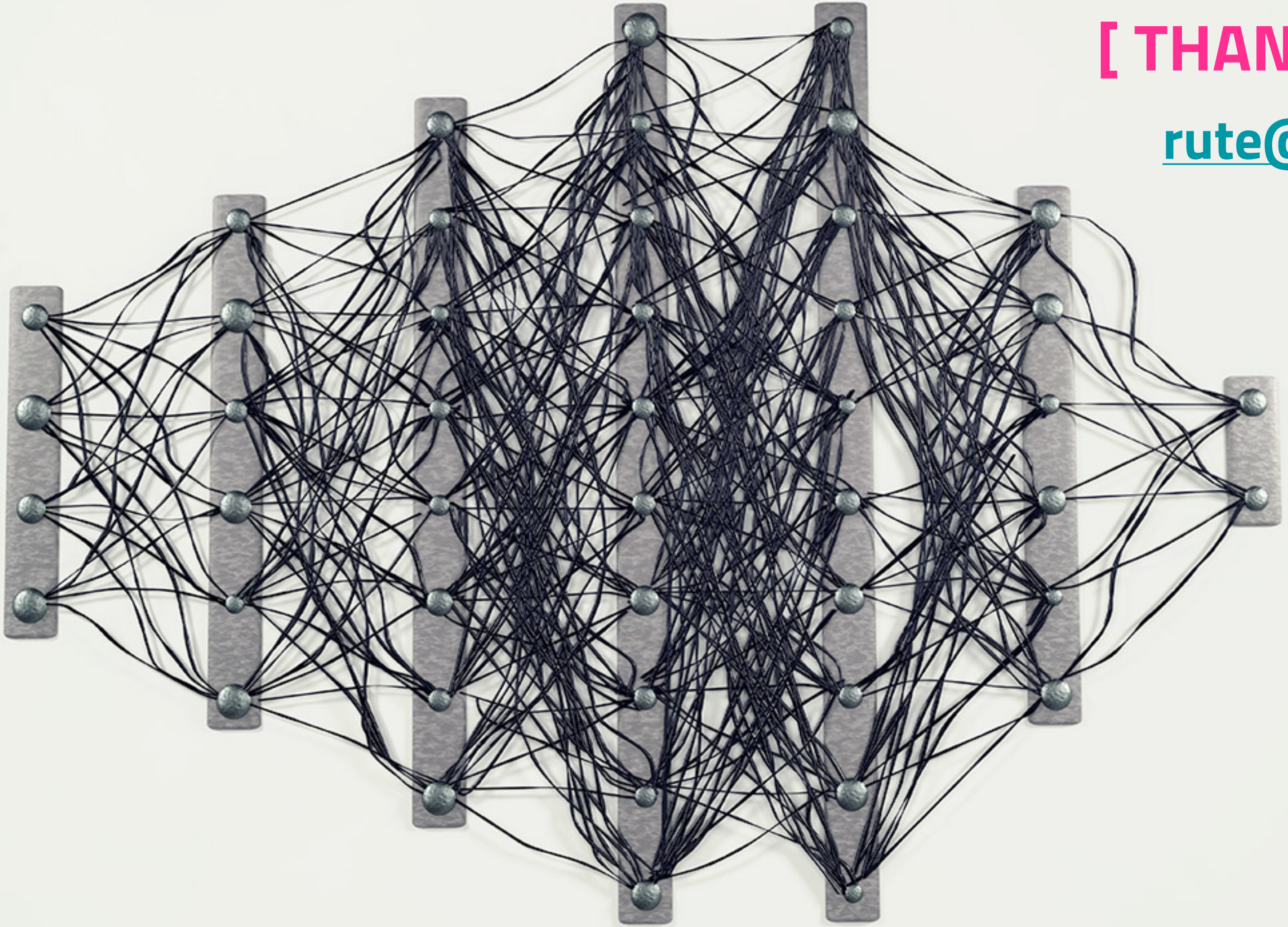
$$\min_{\mathcal{W}} \frac{1}{n} \sum_i ||\mathrm{AE}(\mathbf{x}_i, \mathcal{W}) - \mathbf{x}_i||^2$$



Eur.Phys.J.C 81 (2021)

Summary

- **ML is a universal tool in HEP**, increasing the efficiency of many applications
  - Started well back-ago before Deep Learning revolution
  - Now we use increasingly lower information with deeper and more complex architectures
  - Data representation as images, sets, graphs… to take advantage of the most powerful algorithms
  - Deep Learning is also a key to address HEP’s future challenges (simulation, tracking…)



[THANK YOU]

rute@lip.pt

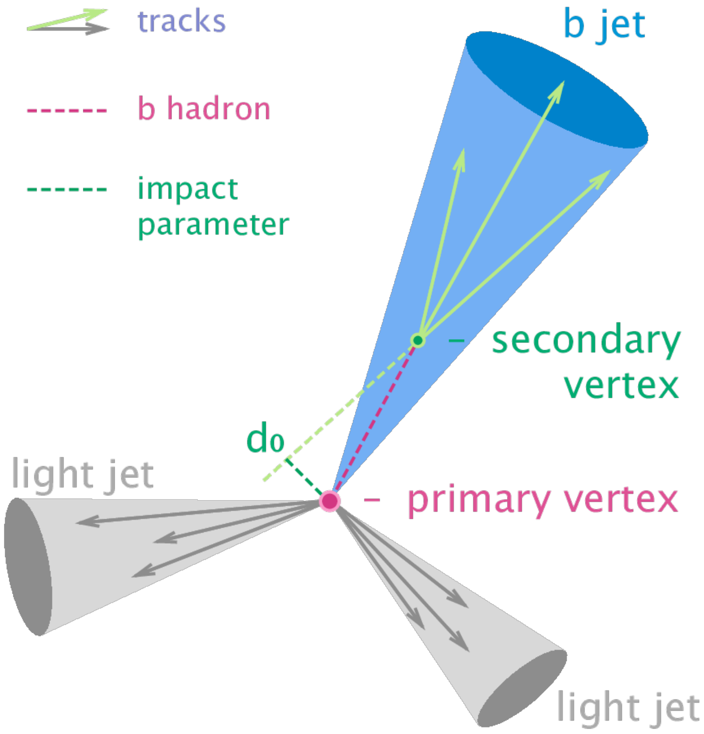
Jet Flavour identification

Essential ingredient for many physics analysis (top, Higgs…)

Per-jet probability of originating from {b, c, uds} quarks

Explore unique characteristics of heavy flavour-jets

- "Large" lifetime of b/c-hadrons (~ps)
- Displaced secondary vertex
- Soft lepton from b/c hadron decay



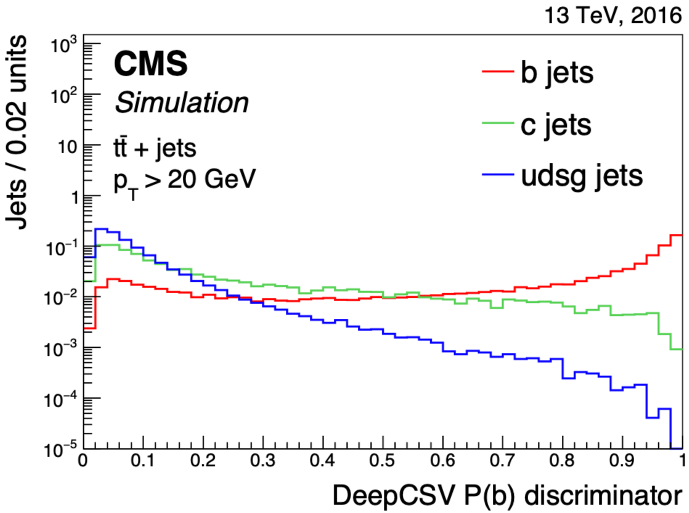
Jet Flavour identification

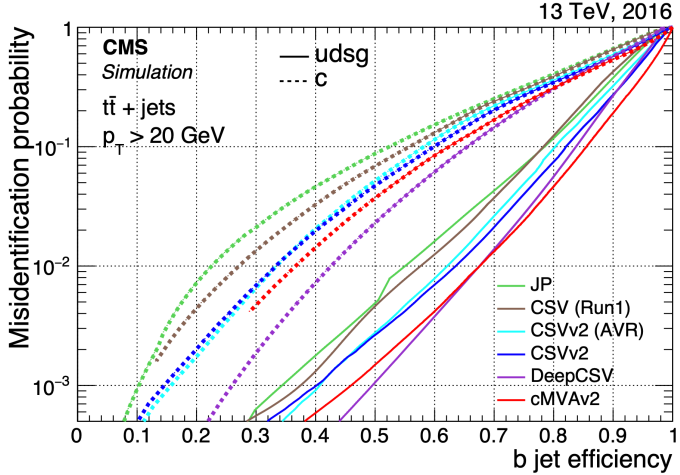
State-of-the-art Deep Learning

New **DeepCSV** (DNN) using same variables of shallow predecessor

- Number of secondary vertices (SV)
- Number of tracks from SV
- SV mass
- Radial distance $\Delta R(\text{track}, \text{jet})$
- Jet p_T, η
- …

Improved efficiency





[1712.07158](https://arxiv.org/abs/1712.07158)

Jet Flavour identification

Deep Sets

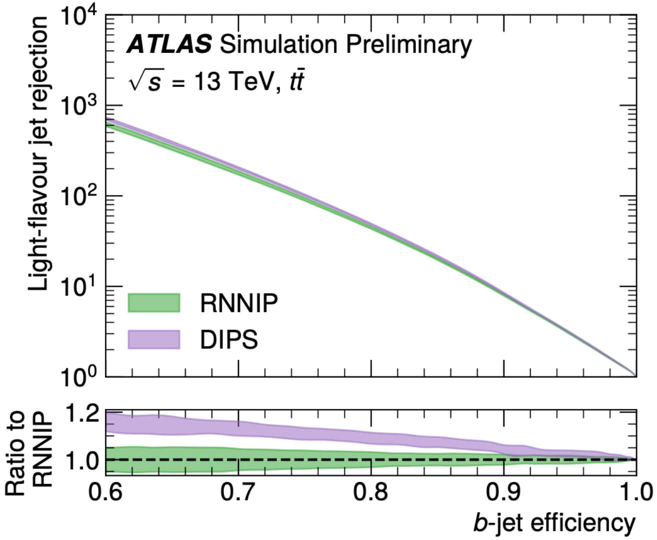
Tagging generally involve a variable number of inputs (tracks)

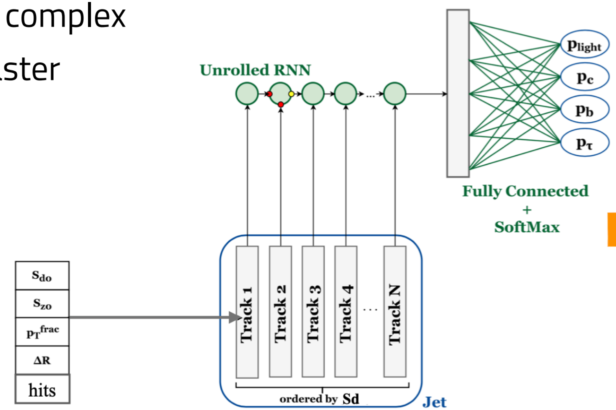
Usually addressed by **Recursive NN**

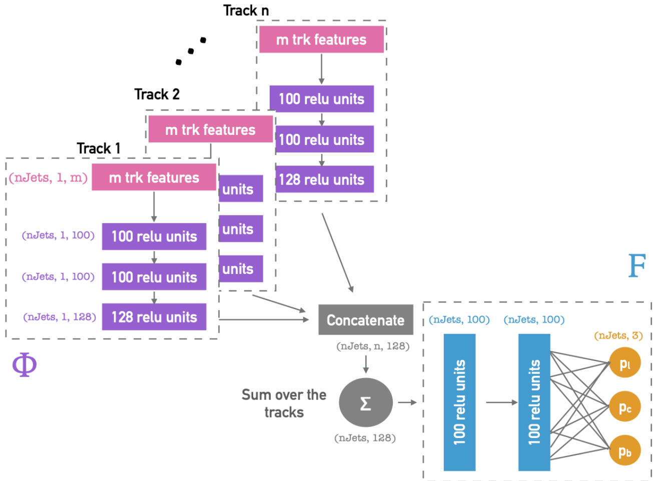
- Natural language processing, order matters (words in sentence)

When order does not matter

- Replace RNN by **DNN + sum**
- Less complex
- 4x faster







ATL-PHYS-PUB-2020-014

Boosted Resonance tagging

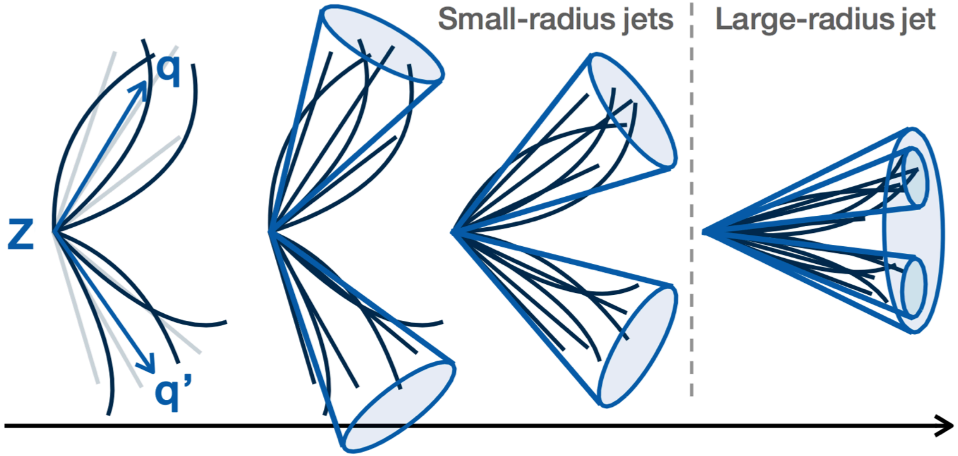
Top, Higgs

Collisions with large energy transfer (Q) are more sensitive to New Physics effects

- Lead to boosted outgoing particles: hadronically decaying resonances are large-jets

Identify the resonant particle

- $t \to Wb \to jjb$ (3 sub-jets, 1 b-sub-jet)
- $H \to bb$ (2 b-sub-jets)
- ...
- Reject non-resonant QCD jets

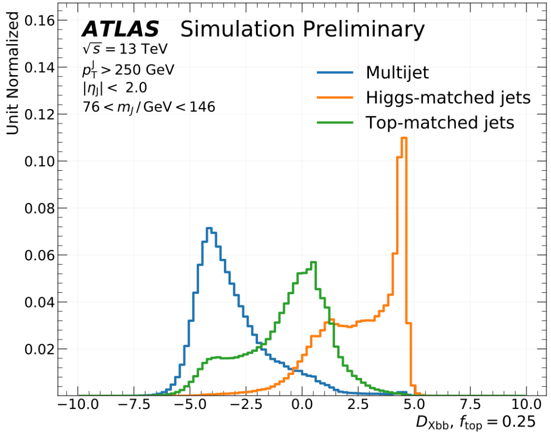


Boosted jets: Increasing transverse momentum, p_T

Boosted Resonance tagging

Xbb tagger

$$D_{\mathrm{Xbb}} = \ln \frac{p_{\mathrm{Higgs}}}{f_{\mathrm{top}} \cdot p_{\mathrm{top}} + (1 - f_{\mathrm{top}}) \cdot p_{\mathrm{multijet}}}$$

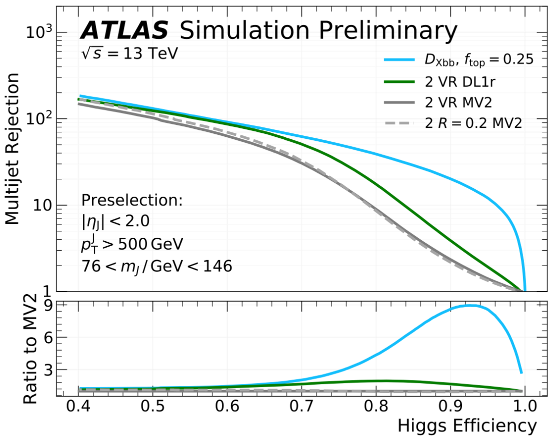


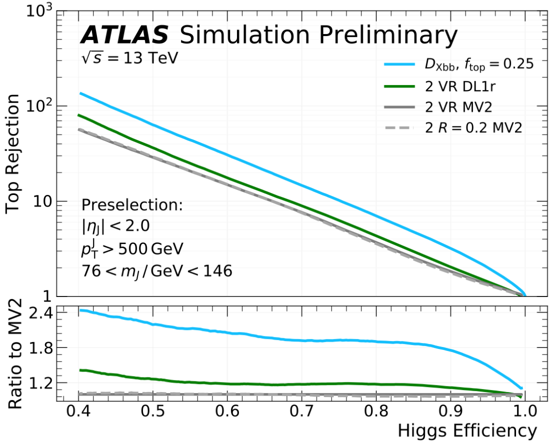
Per-jet probability of being {top,Higgs,QCD}, Multiclass DNN

- B-tagging information from 3 sub-jets
  - Sub-jet probability of being {b,c,uds}
  - Already based on Deep Learning
  - Chaining ML algorithms…

Improvement w.r.t. simple requirement

- 2 b-sub-jets





ATL-PHYS-PUB-2020-019

I. Ochoa LIP seminar

Transferability of DL in Searches for NP

- DNN implemented with Keras using Tensorflow as backend
- Network architecture: Bayesian optimisation using Scikit-Optimize
  - Focus the hyper-parameter tuning where the probability for obtaining the optimal model is larger (depends on past architecture trials)

TABLE I. Hyperparameters used by all DNNs.

Hyperparameter	Value
Hidden Layers	3
Units	352
Unit Activation Function	Selu
Unit Weights Initialiser	LeCun Normal
Dropout Rate	10%
Initial Learning Rate	10^{-3}
Optimizer	Nadam
Maximum Epochs	1000

$$\mathrm{selu}(x) = \lambda \begin{cases} x & \text{if } x > 0 \\ \alpha e^{x} - \alpha & \text{if } x \leqslant 0 \end{cases}$$

[2005.02983](2005.02983)

Continue enhancing Generic Signal Searches

Unsupervised Learning (CWoLa)

Look for new resonances of the form $A \rightarrow B + C$, {B,C} are large-radius jets

- {A,B,C} can be W', Z', graviton,...
- Classifier trained on data only: "signal" enriched sample against background enriched region
- If a real signal exists on data, the DNN will learn to recognise it
- Enhancement of bump hunt in the m_{BC} spectrum