Machine Learning and Big Data for Future Particle Colliders

Nicola De Filippis
Politecnico and INFN Bari

"Data Science Applications in Physics" Winter School in Tirana 2025

Tirana, January 27-30, 2025

Machine learning: definition

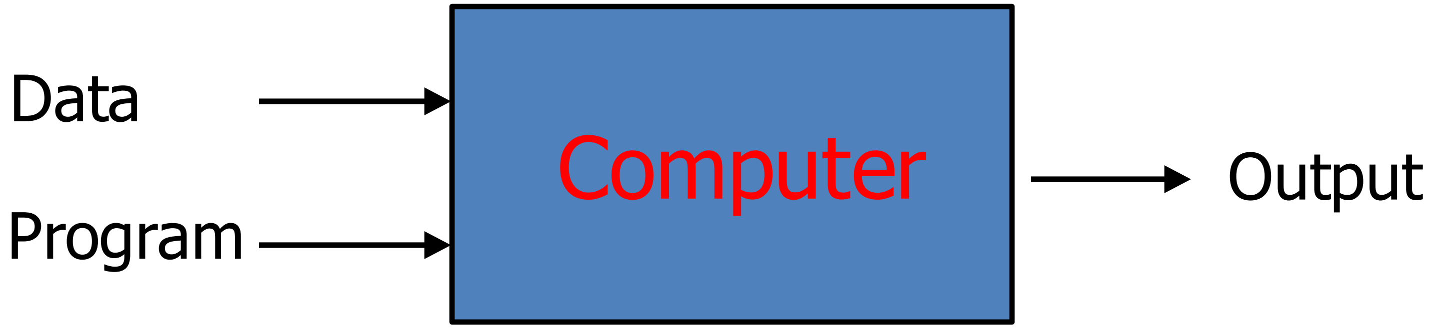
"Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed." -Arthur Samuel (1959)

"Learning is any process by which a system improves performance from experience."

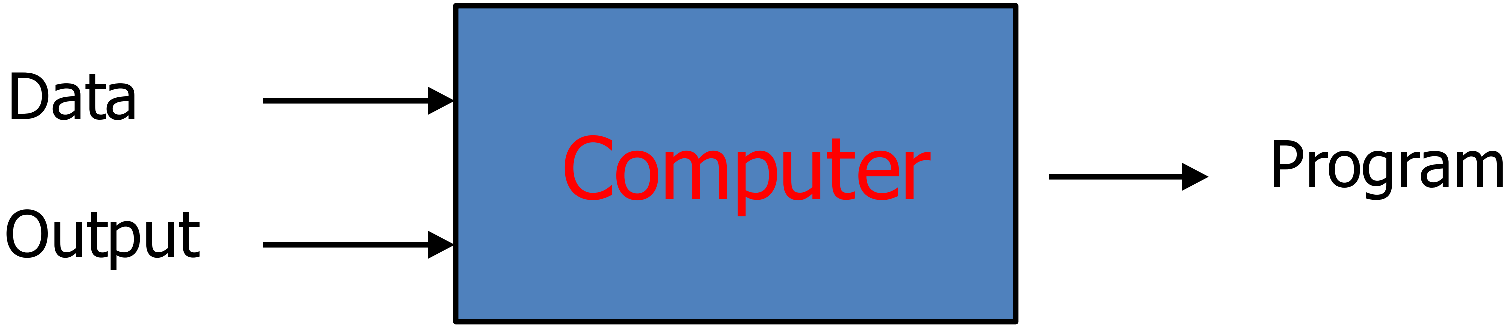
- Herbert Simon

Comparison of different approaches

Traditional programming



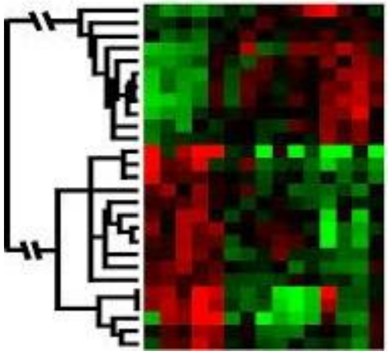
Machine learning



When Do We Use Machine Learning?

ML is used when:

- Human expertise does not exist (navigating on Mars)
- Humans can't explain their expertise (speech recognition)
- Models must be customized (personalized medicine)
- Models are based on huge amounts of data (genomics)
- Fundamental science → HEP



Learning isn't always useful:

- There is no need to "learn" to calculate payroll

More examples of tasks that are best solved by using ML

- Recognizing patterns:
  - Facial identities or facial expressions
  - Handwritten or spoken words
  - Medical images
- Generating patterns:
  - Generating images or motion sequences
- Recognizing anomalies:
  - Unusual credit card transactions
  - Unusual patterns of sensor readings in a nuclear power plant
- Prediction:
  - Future stock prices or currency exchange rates

ML applications

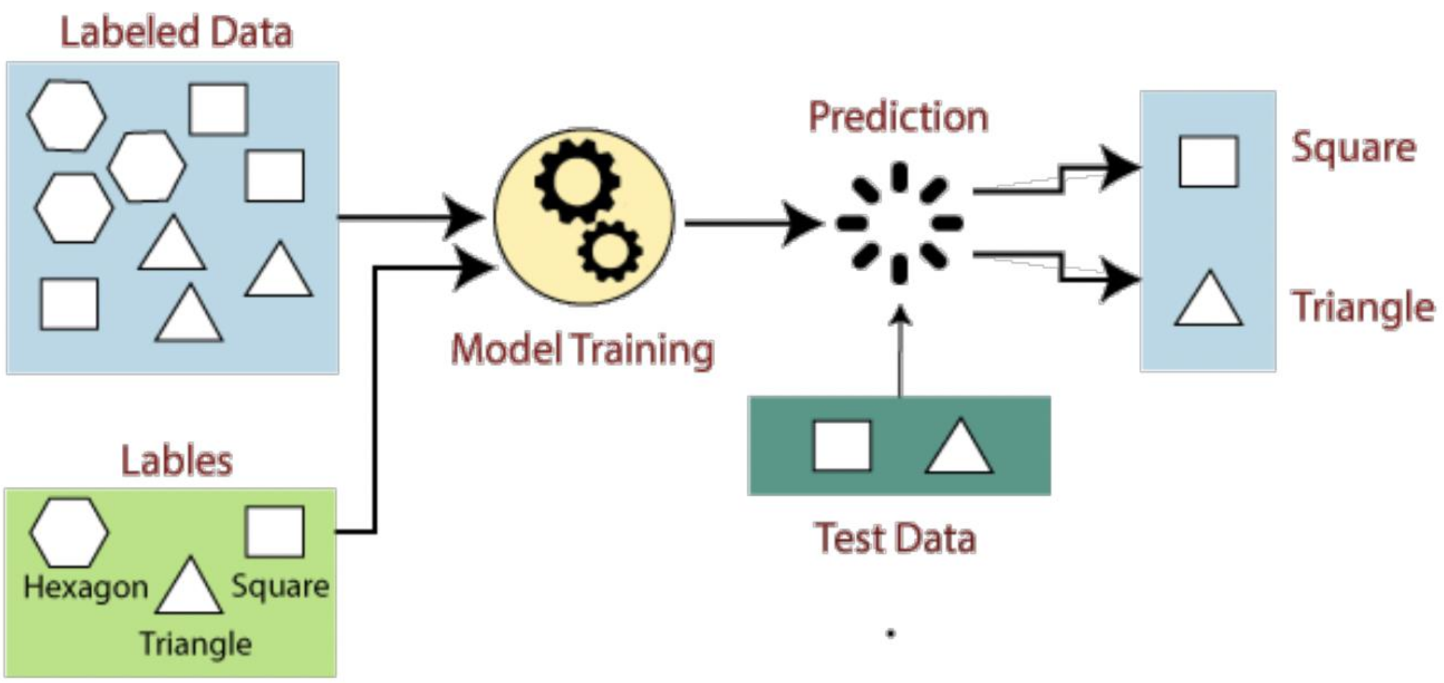
- Web search
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Debugging software
- Fundamental Science → HEP

Type of Learning

- Supervised (inductive) learning
  - Given: training data + desired outputs (labels)
- Unsupervised learning
  - Given: training data (without desired outputs)
- Semi-supervised learning
  - Given: training data + a few desired outputs
- Reinforcement learning
  - Rewards from sequence of actions

Supervised Learning

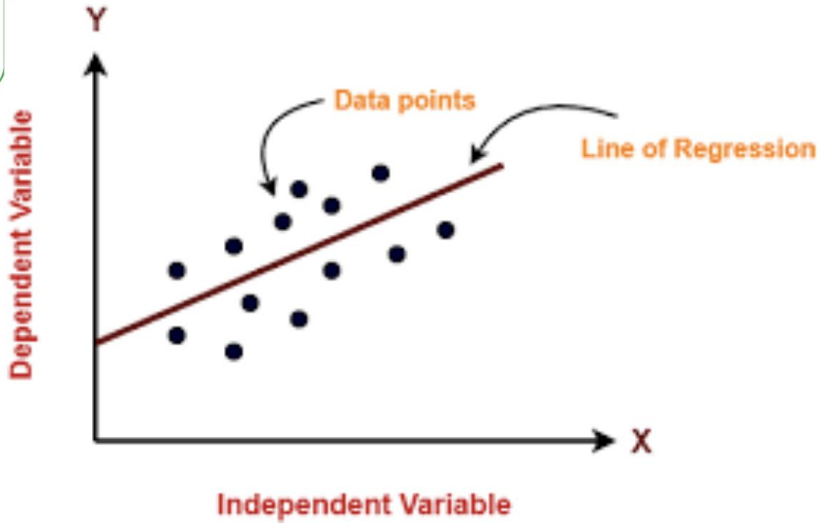
The goal of supervised learning is to learn a model from labeled training data that allows us to make predictions about unseen data.



Supervised Learning: Regression

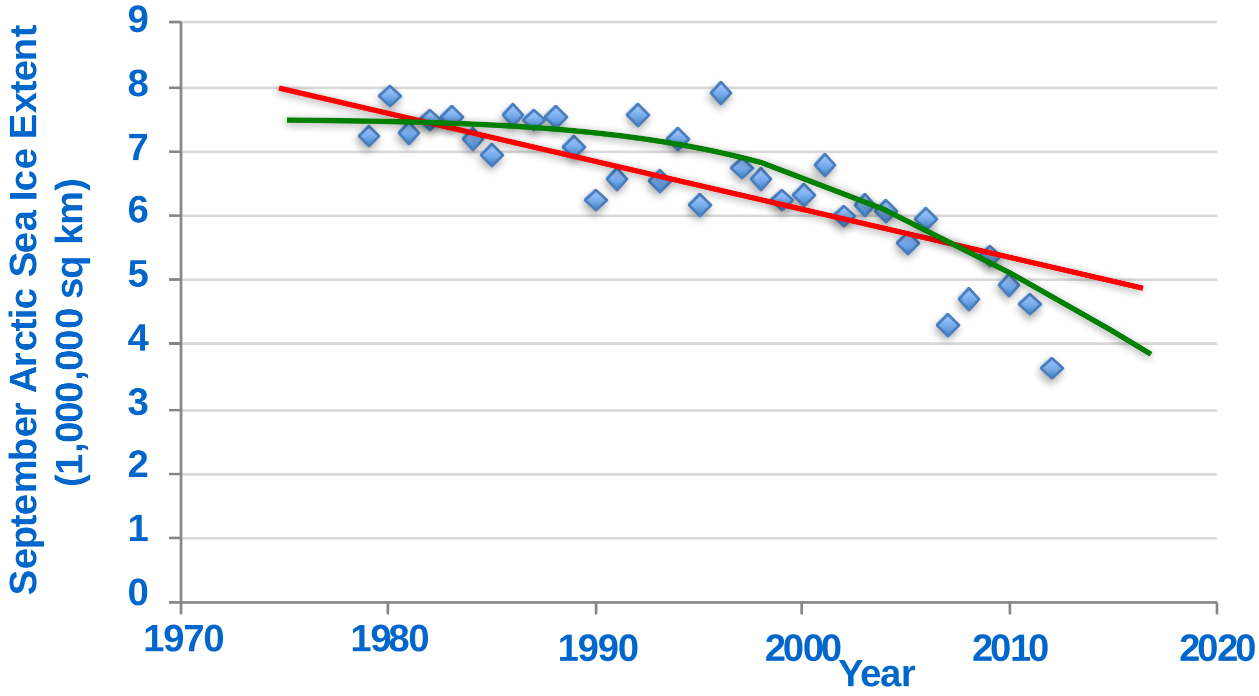
The goal of the regression is the prediction of ***continuous outcomes***

SCORE PREDICTIONS



Supervised Learning: Regression

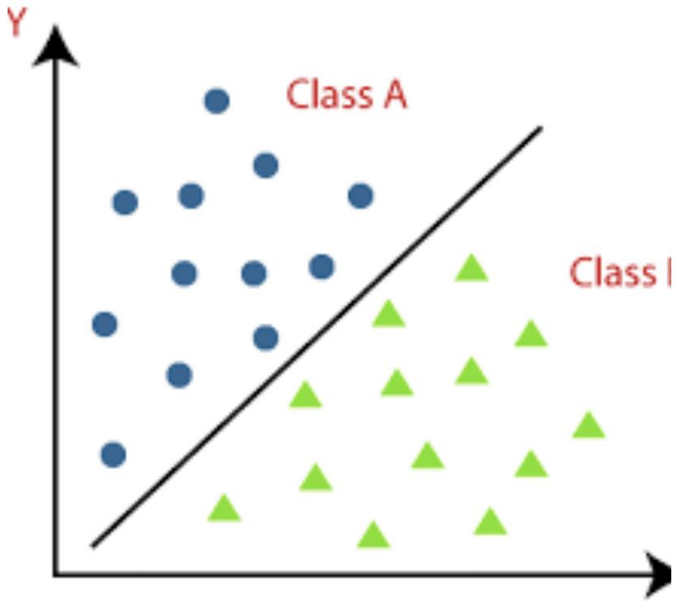
- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is real-valued == regression



Data from G. Witt. Journal of Statistics Education, Volume 21, Number 1 (2013)

Supervised Learning: Classification

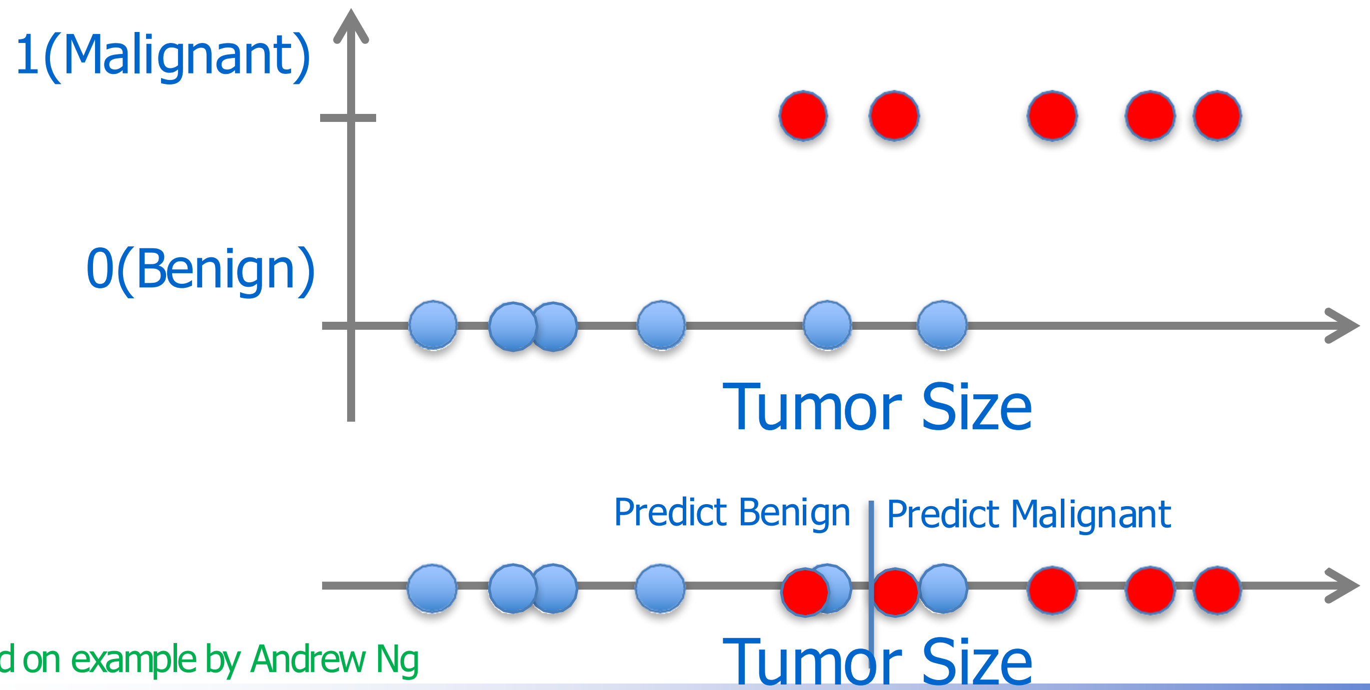
- *The goal of the classification is to predict the categorical class labels of new data based on past observations. Examples are:*
  - ***EMAIL SPAM****: binary classification*
  - ***HANDWRITTEN DIGIT RECOGNITION****: multiple class classification*



Supervised Learning: Classification

- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is categorical == classification

Breast Cancer (Malignant / Benign)



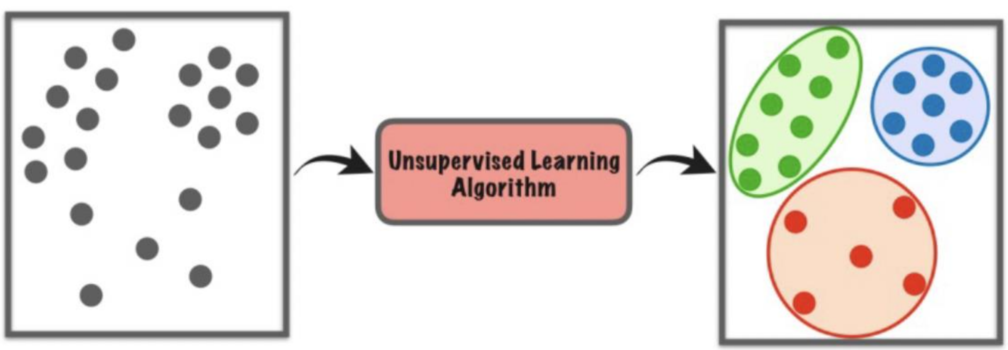
Based on example by Andrew Ng

Unsupervised Learning

The goal of unsupervised learning is to explore the ***structure of data*** *in order to extract meaningful information without the guidance of a known outcome variable or reward function*

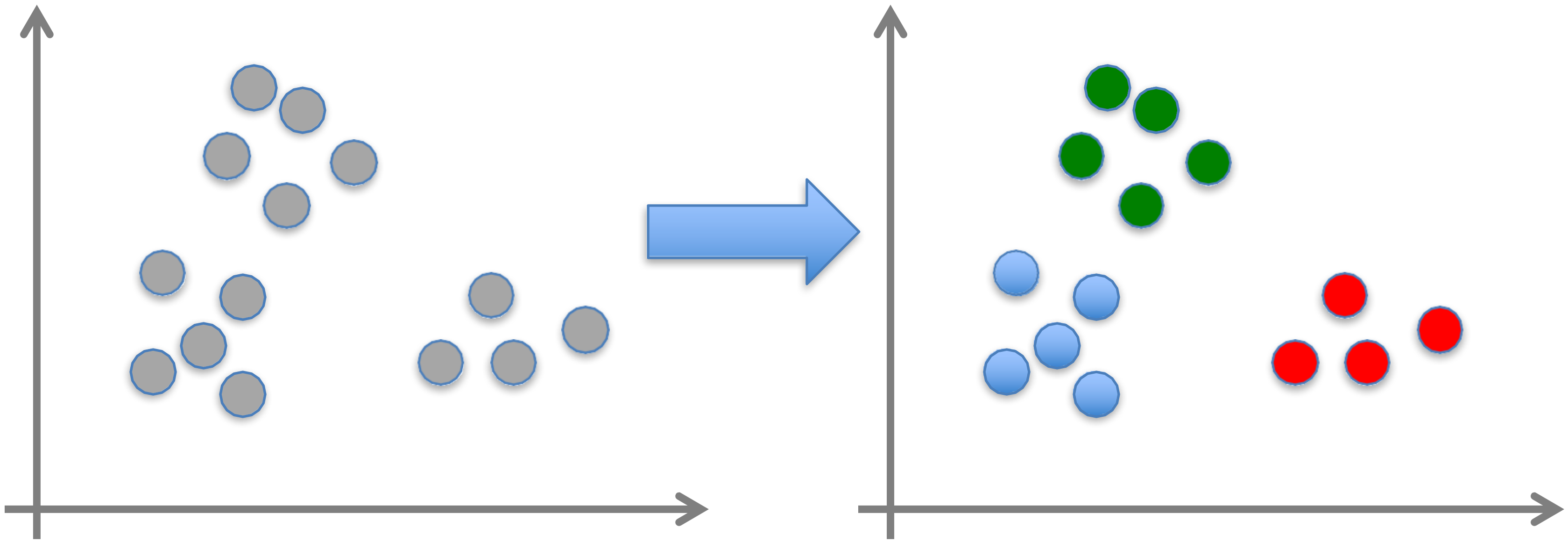
Clustering*: organize information into meaningful subgroups (clusters) without having any prior knowledge of their group memberships*

Dimensionality Reduction



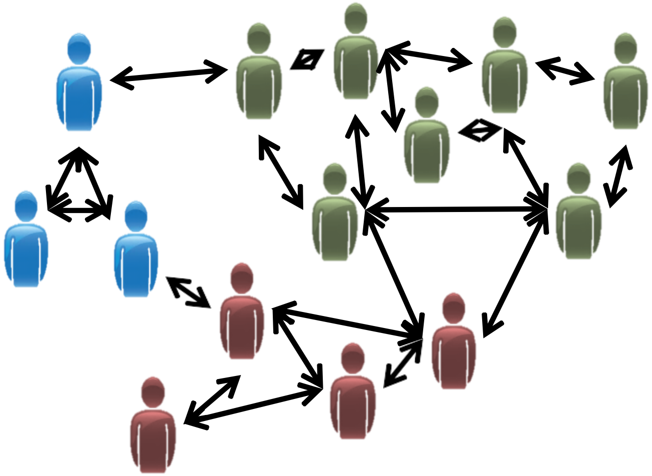
Unsupervised Learning

- Given $x_1, x_2, \ldots, x_n$ (without labels)
- Output hidden structure behind the x's
  - E.g., clustering



Unsupervised Learning

Organize computing clusters



Social network analysis

Market segmentation

Image credit: NASA/JPL-Caltech/E. Churchwell (Univ. of Wisconsin, Madison)

Astronomical data analysis

Unsupervised Learning

- Independent component analysis – separate a combined signal into its original sources

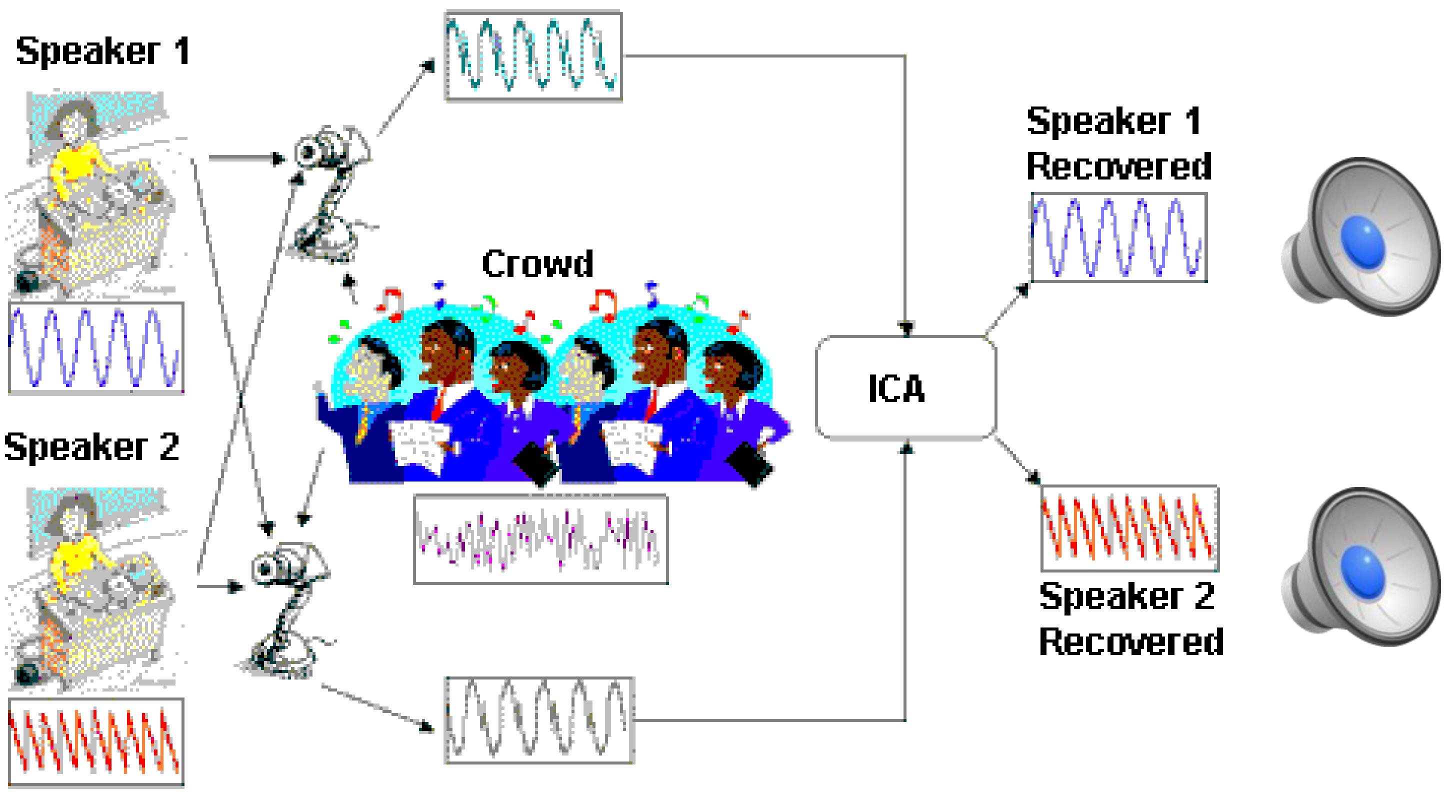
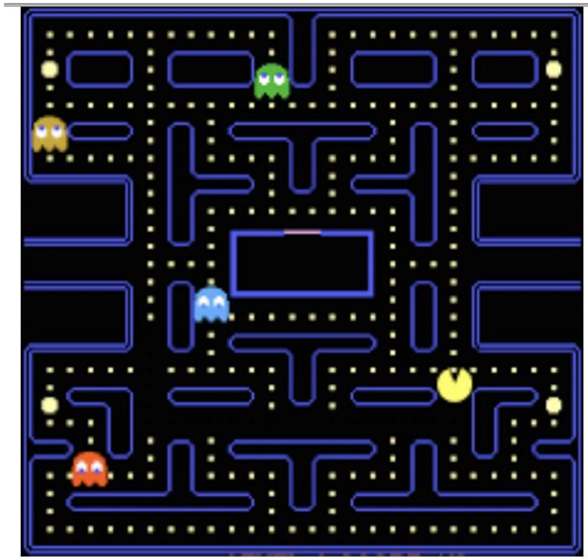
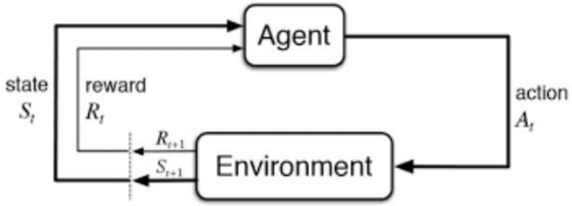


Image credit: statsoft.com Audio from http://www.ism.ac.jp/~shiro/research/blindsep.html

Reinforcement Learning





The goal of reinforcement learning is the development of a system which improves by interacting with the environment

→

PACMAN GAME

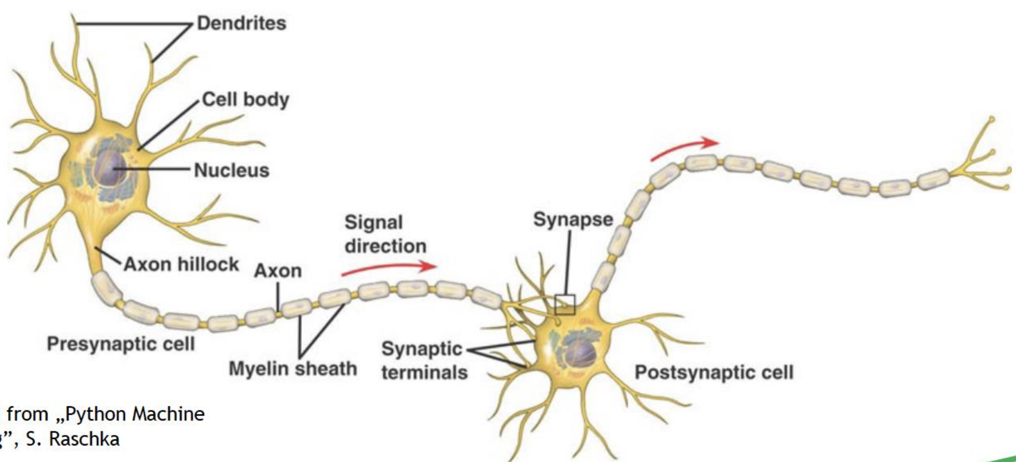
Reinforcement Learning

- Given a sequence of states and actions with (delayed) rewards, output a policy
  - Policy is a mapping from states → actions that tells you what to do in a given state

- Examples:
  - Credit assignment problem
  - Game playing
  - Robot in a maze
  - Balance a pole on your hand

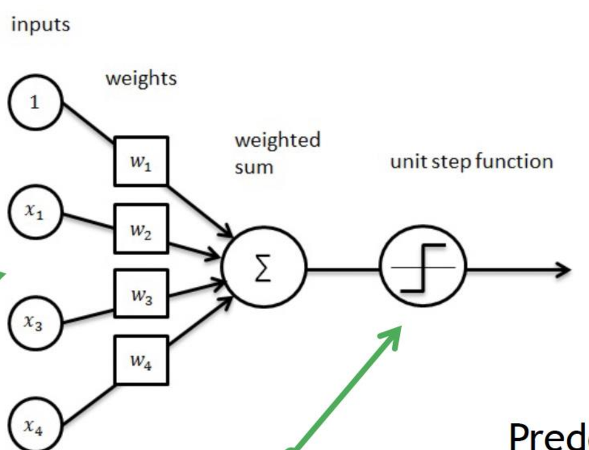
Learning algorithm and Artificial neuron or Perceptron

- For our purpose we define a learning algorithm (LA) as a composite entity including:
  - a data set, for which we search for patterns
  - a model (for our discussion here, this will be represented by weights)
  - an optimisation algorithm (a recipe to adjust/change weights)
  - a loss function
  - LA is able to learn based on the data that is „given" to it
  - To be able to describe the learning process in quantitative way we define, on top of the previous notions, Experience, Class of Tasks and Performance Metric



Adapted from „Python Machine Learning", S. Raschka

- **1943** with McCullock-Pitts **neuron model**
- **Motivated by biological studies**



- **Perceptron equation**

$$z^{(i)} = w_1 x_1^{(i)} + w_2 x_2^{(i)} + \cdots + w_k x_k^{(i)} = \sum_{j=1}^{j=k} w_j x_j^{(i)} = \vec{w}^T \vec{x}^{(i)}$$

$$\phi(z) = \begin{cases} +1 \ if \ z \geq \theta \\ -1 \ if \ z < \theta \end{cases}$$

Predefined threshold

The algorithm

- The perceptron algorithm, then goes like that:
  - **Initialise the weights vector to 0 or „something small”**
  - **For each training data sample $\vec{x}^{(i)}$ do:**
    - **Get the output value (class label) $\tilde{y}^{(i)}$, using the unit step function**
    - **Update the weights accordingly (update concerns all the weights in one go)**
- We can write

$$w_j = w_j + \Delta w_j$$

$$\boldsymbol{\Delta w_j = \eta \cdot \left(y^{(i)} - \tilde{y}^{(i)}\right) \cdot x_j^{(i)}}$$

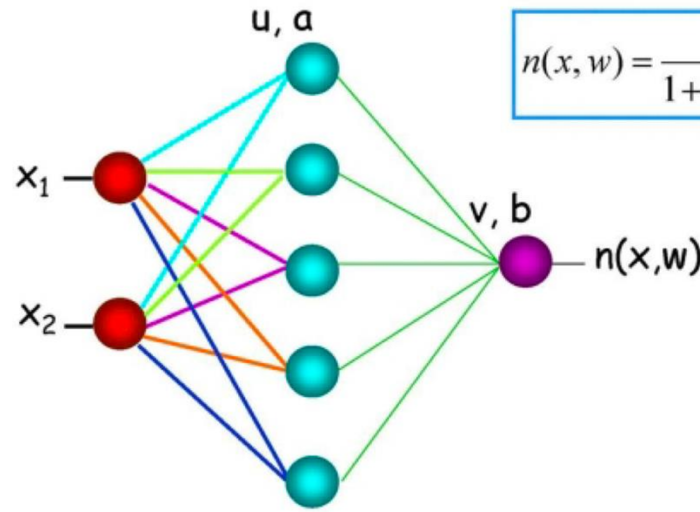
- The second formula is called **perceptron learning rule**, and the $\boldsymbol{\eta}$ is called the learning rate (just a number between 0 and 1)

Feed-forward neural networks

- Learning can be viewed as using direct or indirect experience to approximate a chosen target function.
- Function approximation can be viewed as a search through a space of hypotheses (representations of functions) for one that best fits a set of training data.
- Different learning methods assume different hypothesis spaces (representation languages) and/or employ different search techniques.

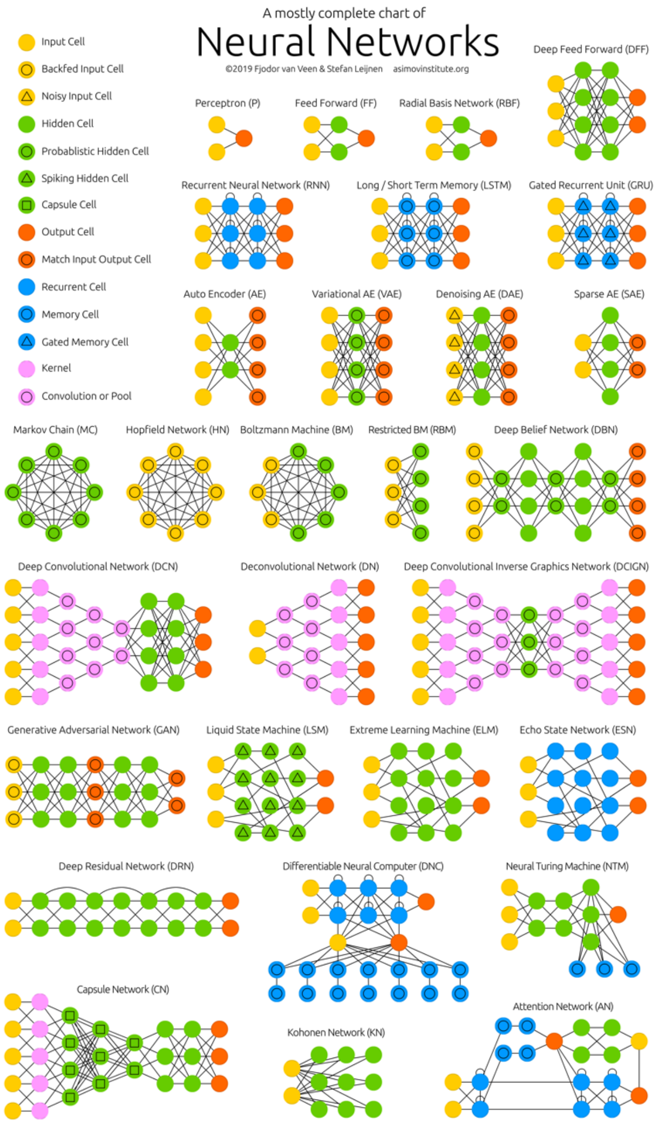
$$f(x,w) = b + \sum_{j=1}^{H} v_j \tanh\left[a_j + \sum_{i=1}^{P} u_{ij} x_i \right]$$

$$n(x,w) = \frac{1}{1+\exp[-f(x,w)]}$$



Most of ML is concerned with how to find the weights such that your NN produces accurate opinions

The neural network zoo



Neural network architectures popping up every now and then, it's hard to keep track of them all

This is cheat sheet containing many of the NN architectures.

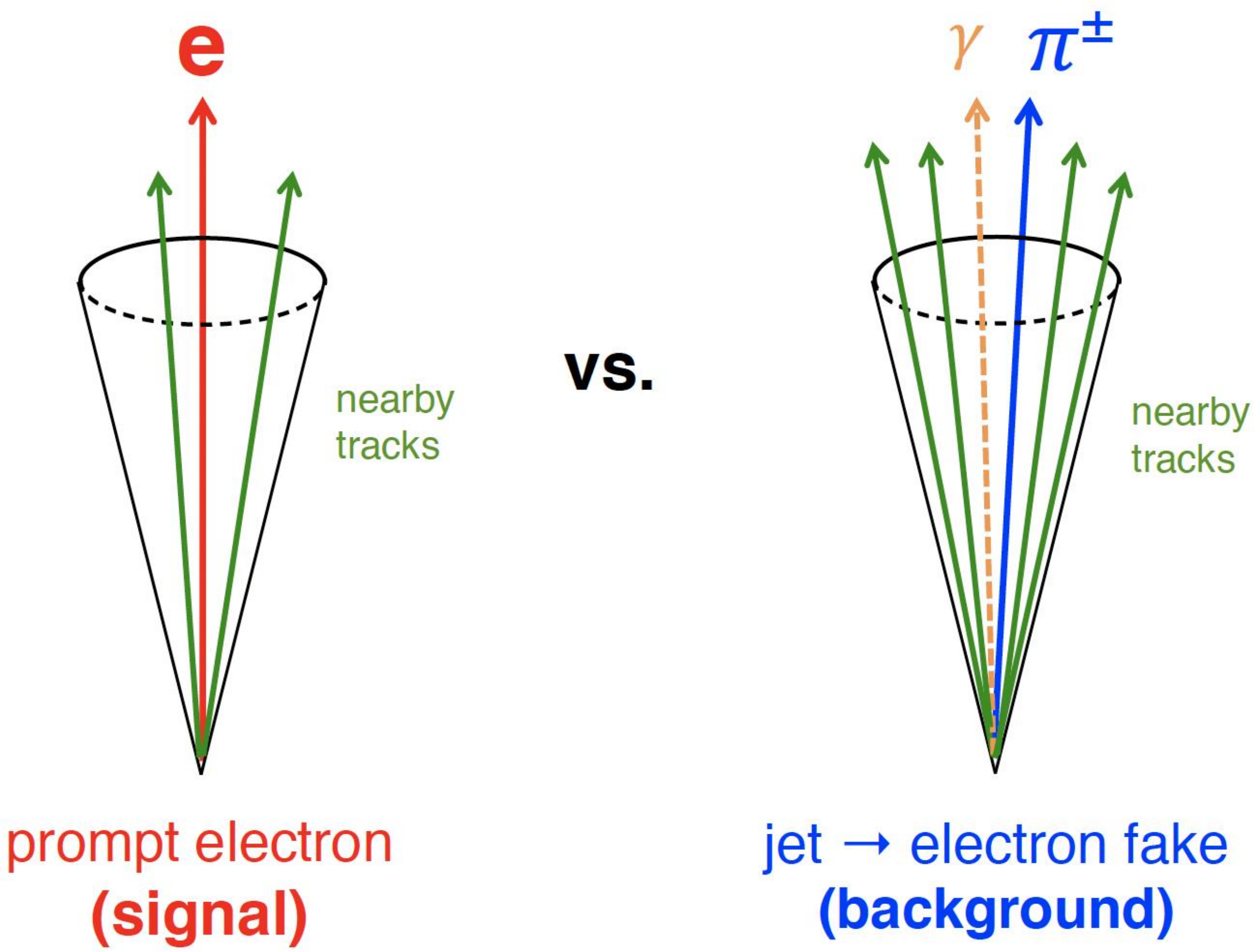
https://www.asimovinstitute.org/neural-network-zoo/

Machine Learning for HEP

Machine learning use cases at HEP colliders

- Fast simulation
- Tracking with unsupervised learning
- Jet classification
- Particle ID
- Event-based classification
- Physics analysis

Intro: Classification at Colliders



How do we identify electrons at LHC?

Classification techniques at Colliders

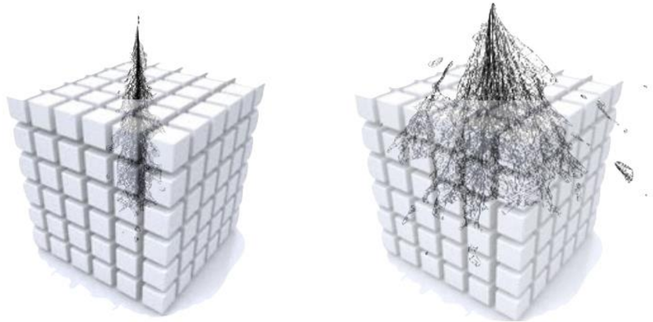
1. **Cut-based selection**
   - Apply requirements on human-designed **features**

machine learning

2. **Multi-Variate Algorithms (MVA)**
   - Combine features using *neural networks, boosted decision trees, likelihoods, etc.*
   - **Exploit *correlations* between features**

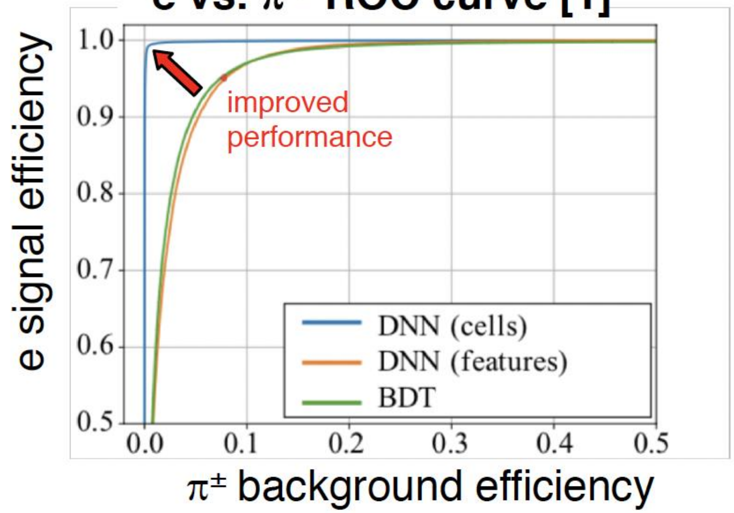
3. **Deep Learning**
   - Feed *low-level data* (e.g. calorimeter cells) directly to **deep neural networks**
   - **Potential to exploit *information not contained in features***

single particle showers in a high-granularity 3D calorimeter



electromagnetic shower | hadronic shower

e vs. $\pi^\pm$ ROC curve [1]



[1] BH, Farbin, Khattak, Pacela, Pierini, Vlimant, Spiropulu, Wei, Proceedings of the Deep Learning for Physical Sciences Workshop at Neural Information and Processing Systems (NIPS17)

Example: in-painting with Deep Learning

corrupted image

deep learning

"in-painted" image
using deep neural networks [2]

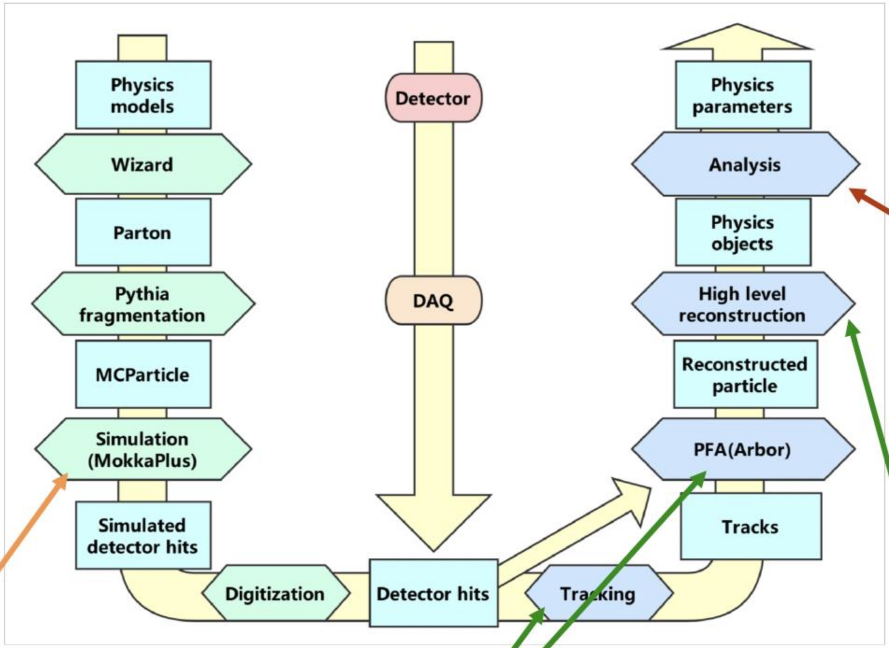
Use Cases at Colliders

MC data physics!

1. Generation
of truth information

2. Simulation
of detector response

generative models
(e.g. calorimeter showers)



4. Analysis
of physics objects

event classification
(e.g. ttH vs. tt+bb)

event regression
(e.g. M_{Higgs})

3. Reconstruction
of physics objects

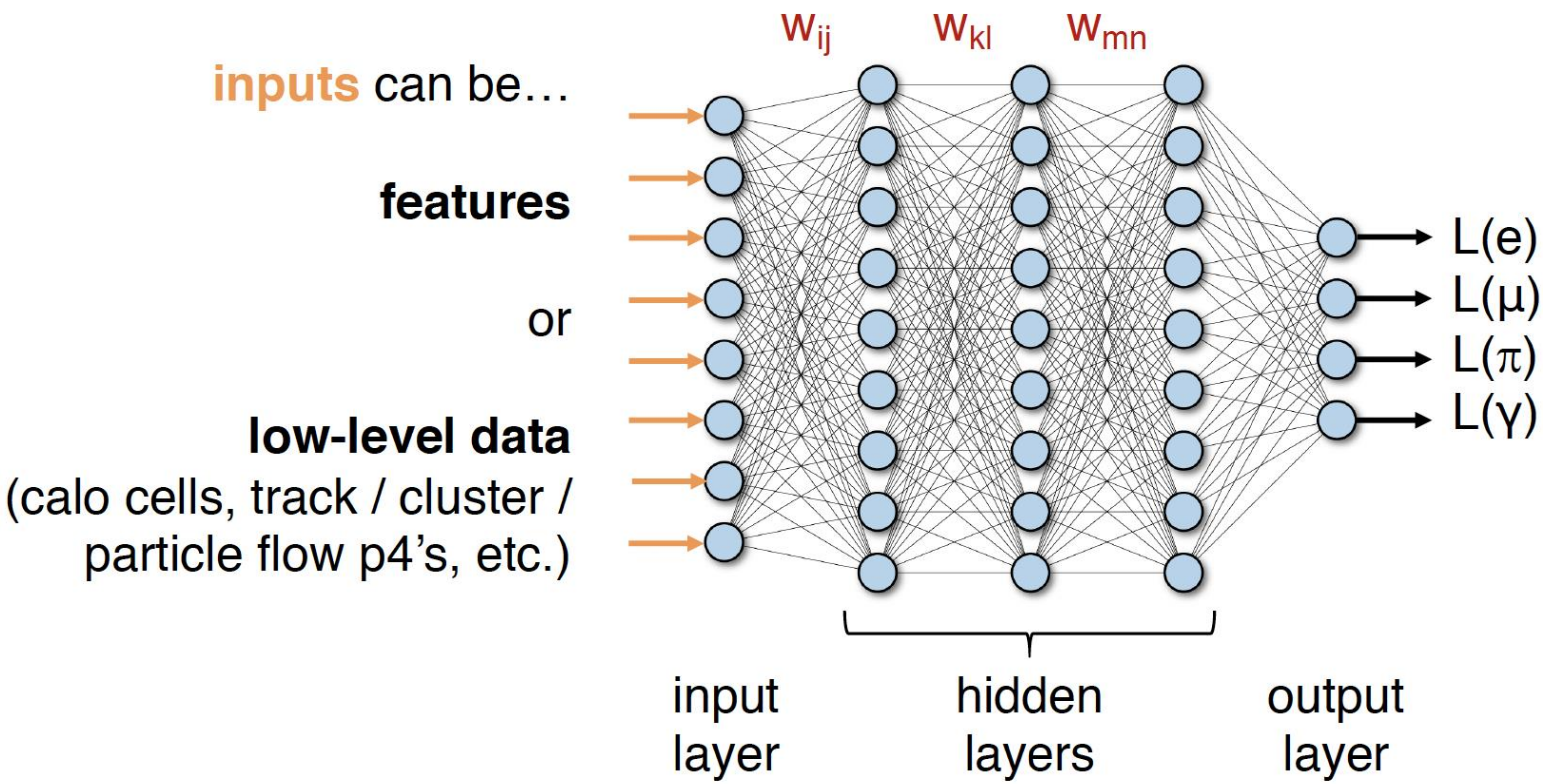
object classification
(e.g. particle ID, b-tagging)

object regression
(e.g. E, θ, ϕ)

unsupervised classification
(e.g. tracking, clustering, track-cluster matching)

Neural Network Architectures

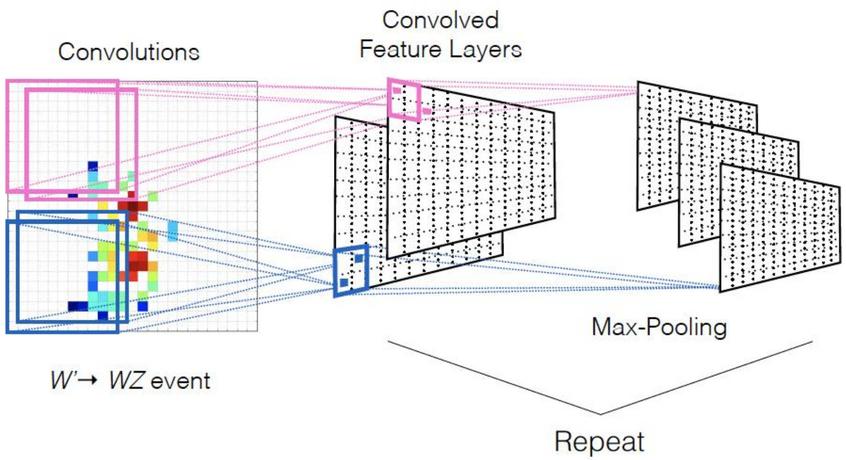
- **Fully-Connected Networks (FCN)**
  - Multiple layers of **fully inter-connected neurons** with variable weights
  - Structure-agnostic → widely applicable

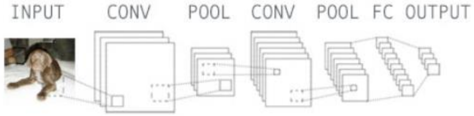


Neural Network Architectures

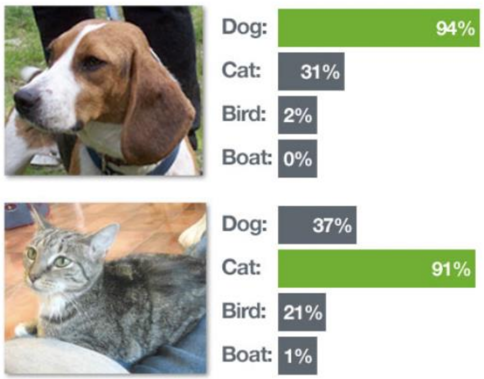
- **Fully-Connected Networks (FCN)**
  - Multiple layers of **fully inter-connected neurons** with variable weights
  - Structure-agnostic → widely applicable

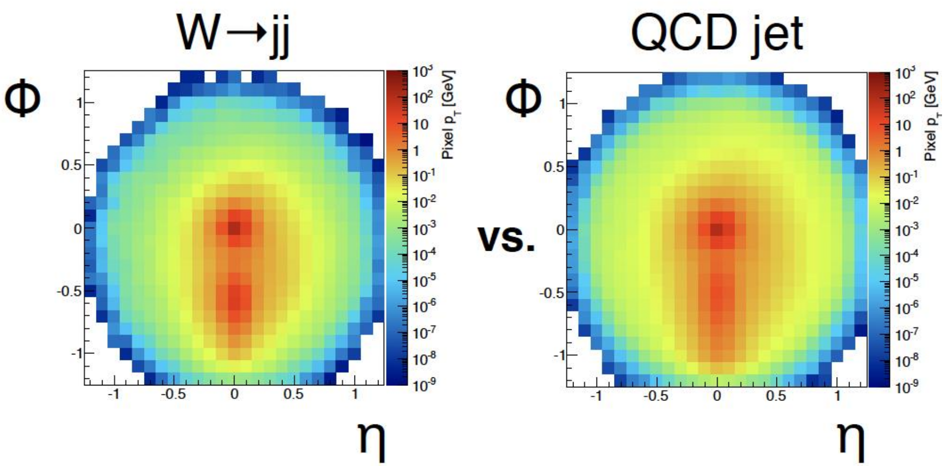
- **Convolutional Neural Networks (CNN)**
  - Specialized layers ("convolutional filters") identify structures at different scales
  - **Computer vision / imaging** applications
  - Assumes **fixed-length** input data





exploits extensive computer vision R&D

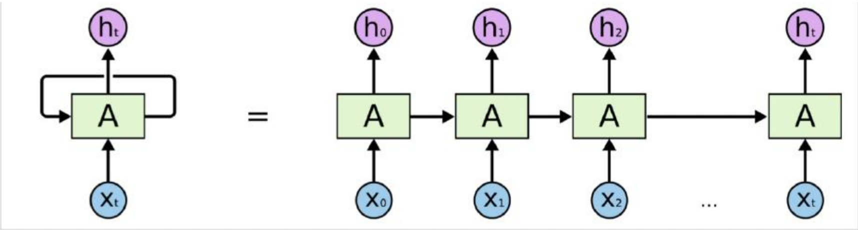


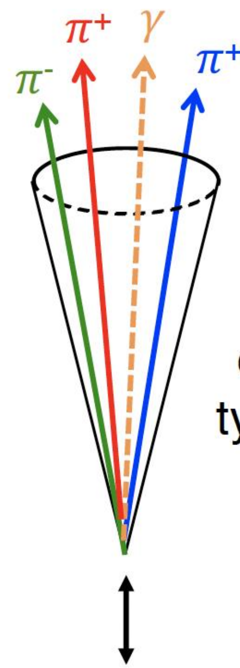


[1] de Oliveira, Kagan, Mackey, Nachmann, Schwartzman, "Jet Images – Deep Learning Edition", JHEP07 (2016) 069

Neural Network Architectures

- **Fully-Connected Networks (FCN)**
  - Multiple layers of **fully inter-connected neurons** with variable weights
  - Structure-agnostic → widely applicable
- **Convolutional Neural Networks (CNN)**
  - Specialized layers ("convolutional filters") identify structures at different scales
  - **Computer vision / imaging** applications
  - Assumes **fixed-length** input data
- **Recurrent Neural Networks (RNN)**
  - Cyclical structures allow for **variable-length** input data
    - ➢ e.g. Particle Flow Candidate p4's
  - **Language processing** applications





jet ⟷ sentence
constituents ⟷ words
type, p_T, η, Φ ⟷ letters

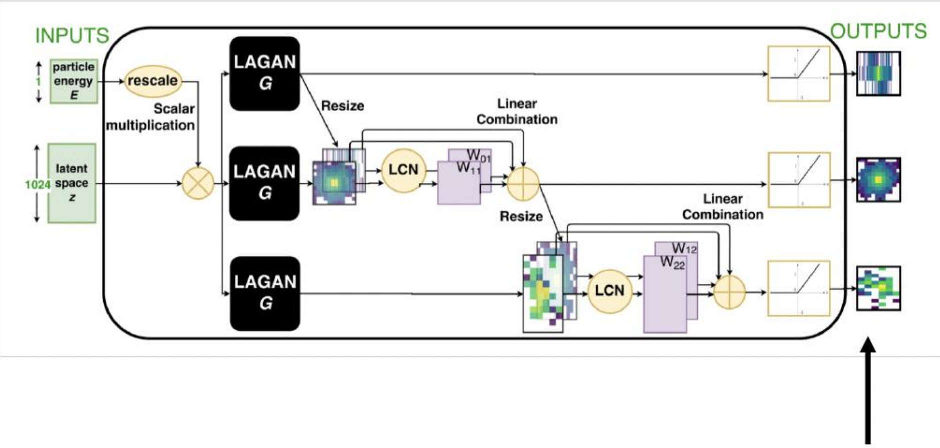
"pm_pt3.5_eta1.1_phi0.2 pp_pt5.6_eta0.3_phi1.8 g_pt10.5_eta1.4_phi0.3 pp_pt3.5_eta1.1_phi1.2."

exploits extensive language processing and translation R&D (e.g. google translate)

Louppe, Cho, Becot, Cranmer, QCD-Aware RNNs for Jet Physics, 1702.00748
Cheng, RNNs for Quark/Gluon Tagging, CSBS (2018) 2:3
ATLAS, b-tagging with RNNs, ATL-PHYS-PUB-2017-003

Neural Network Architectures

- **Fully-Connected Networks (FCN)**
  - Multiple layers of **fully inter-connected neurons** with variable weights
  - Structure-agnostic → widely applicable
- **Convolutional Neural Networks (CNN)**
  - Specialized layers ("convolutional filters") identify structures at different scales
  - **Computer vision / imaging** applications
  - Assumes **fixed-length** input data
- **Recurrent Neural Networks (RNN)**
  - Cyclical structures allow for **variable-length** input data
    - e.g. Particle Flow Candidate p4's
  - **Language processing** applications
- **Generative Adversarial Networks (GAN)**
  - Generate ensembles of pseudo-data
  - **Fast simulation** applications



generated output images (for 3 ATLAS ECAL layers)

Paganini, de Oliveira, Nachman, CaloGAN for 3D particle showers, PRD 97, 014021 (2018)

MC Use Cases at Colliders

- **Fully-Connected Networks (FCN)**
  - Multiple layers of **fully inter-connected neurons** with variable weights
  - Structure-agnostic → widely applicable
- **Convolutional Neural Networks (CNN)**
  - Specialized layers ("convolutional filters") identify structures at different scales
  - **Computer vision / imaging** applications
  - Assumes **fixed-length** input data
- **Recurrent Neural Networks (RNN)**
  - Cyclical structures allow for **variable-length** input data
    - e.g. Particle Flow Candidate p4's
  - **Language processing** applications
- **Generative Adversarial Networks (GAN)**
  - Generate ensembles of pseudo-data
  - **Fast simulation** applications

classification

- objects: jet classification, particle ID, etc.
- events: $t\bar{t}H(b\bar{b})$ vs. $t\bar{t}+b\bar{b}$, SUSY vs. $t\bar{t}$, etc.
- "supervised" (labeled data) or "unsupervised"

measurements with **regression**

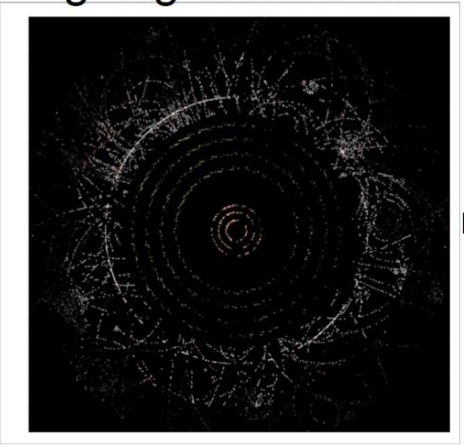
- objects: jet and lepton energies and angles
- events: total / hadronic / missing energy, m_H

fast simulation

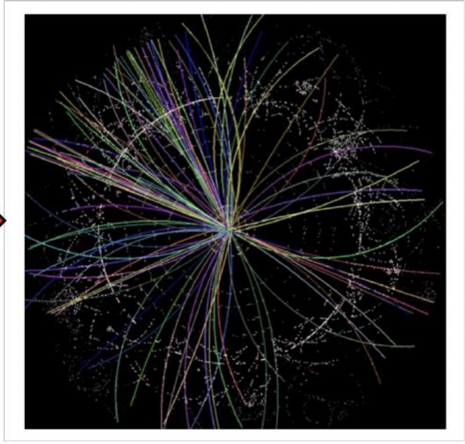
e.g. particle showers in calorimeters

Tracking with ML

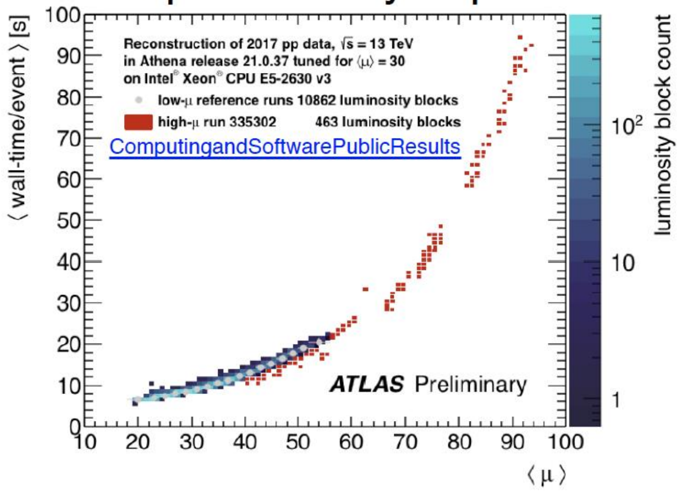
going from hits…



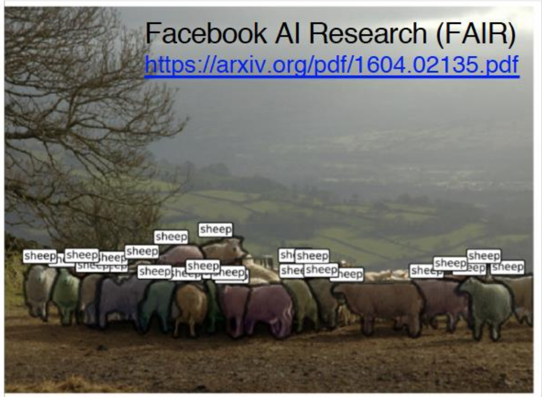
to tracks…

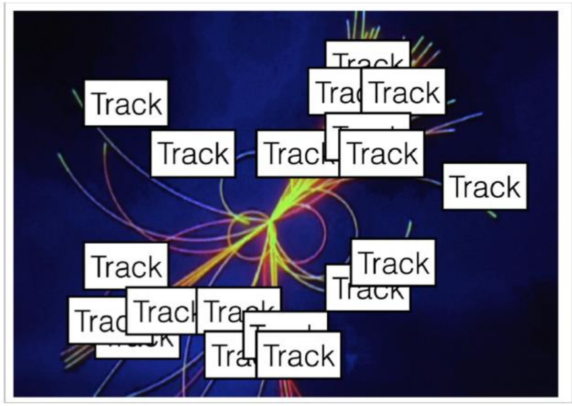


is computationally expensive:



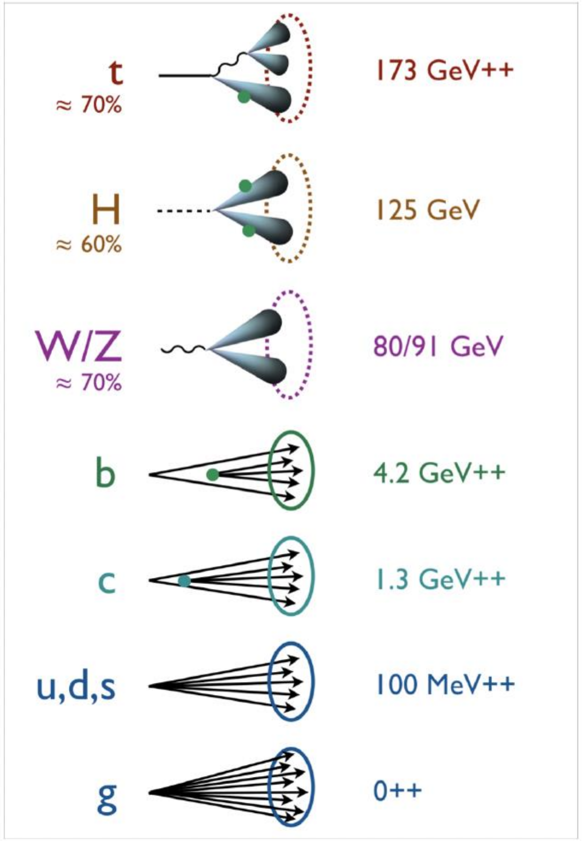
- Major challenge for HL-LHC and future hadron colliders!
- Can leverage **unsupervised learning** techniques to group hits into tracks
- Subject of TrackML challenge



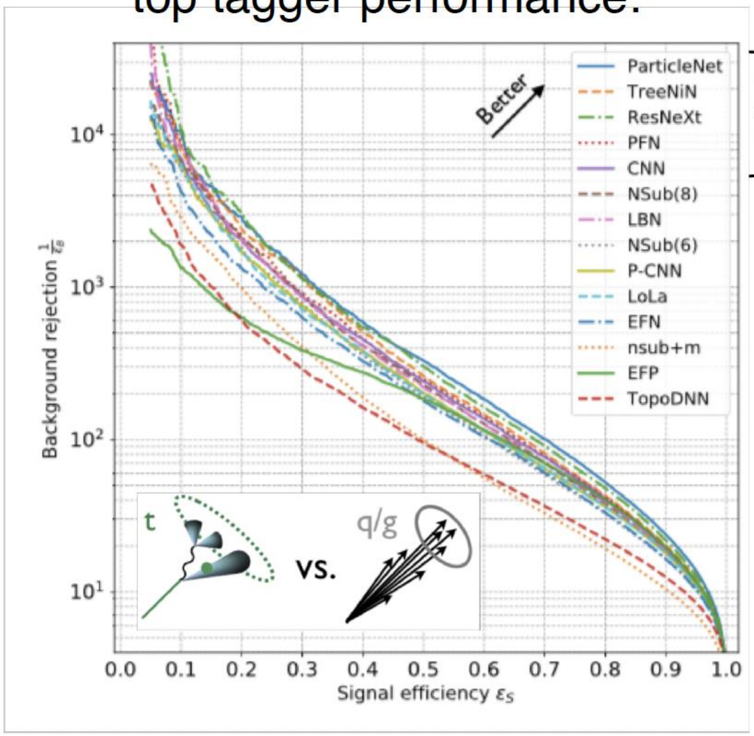


Jet classification with ML

++ = mass from QCD radiation



top tagger performance:

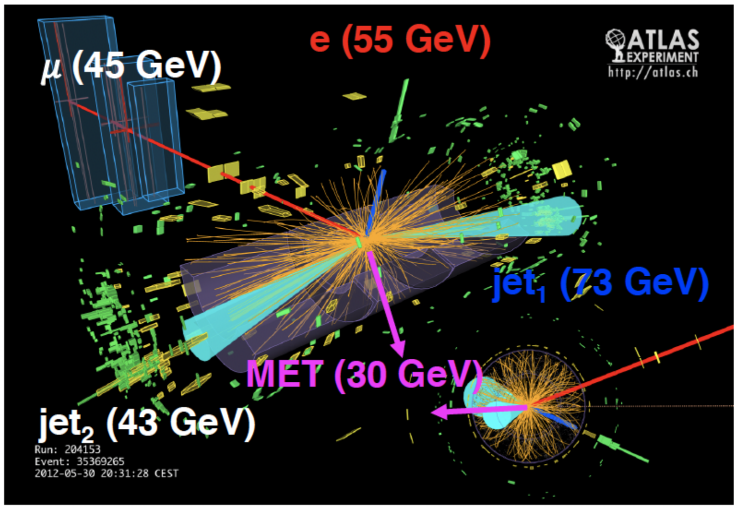


deep learning

[1] from slides by Jessie Thaler
see also recent reviews:
Larkoski, Moult, Nachman, 1709.04464,
Marzani, Soyez, Spannowsky, 1901.10342

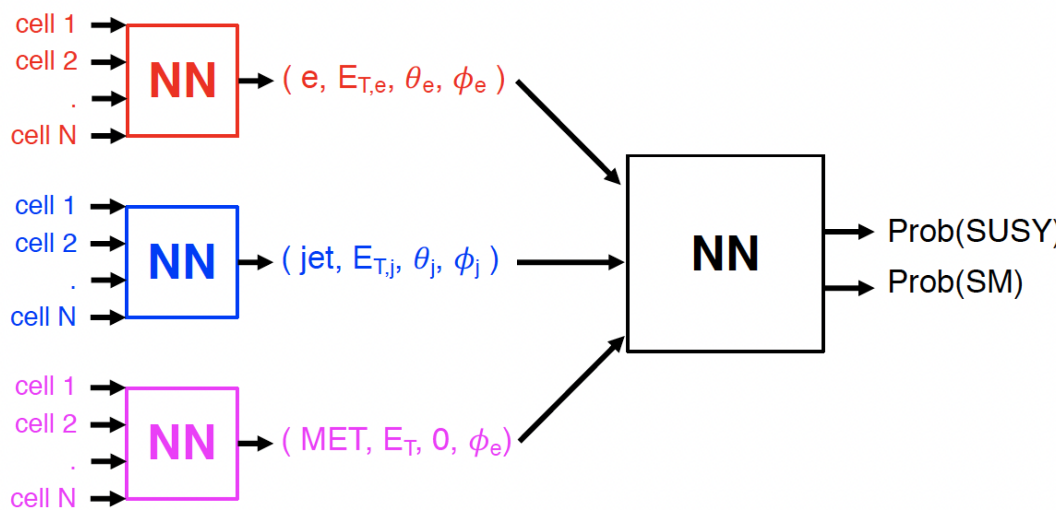
- Deep learning approach often provides best performance for jet classification tasks

Strategy for ML event classification



object classification & regression

event categorization

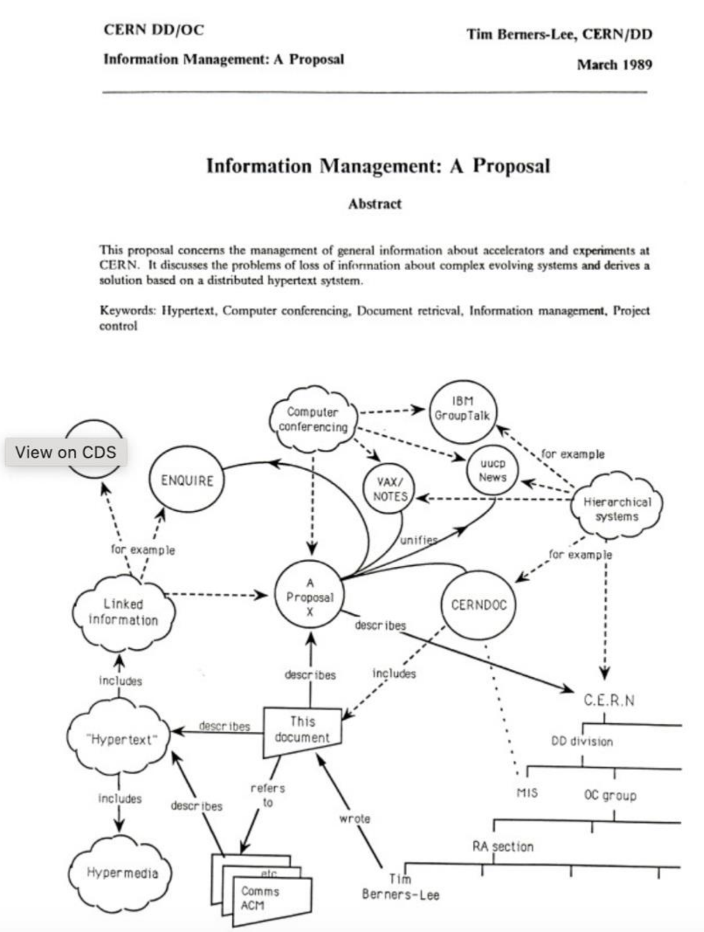


- Factorize the problem: **object tagging** + **event classification**
  - Use **cells** to *classify type* and *measure p4's* of physics **objects** (e, μ, τ, γ, j, MET)
  - Use **object types and p4's** to *categorize* **events** (e.g. SM vs. SUSY) with e.g. RNNs

Big data for HEP

History of computing - www

How the Web began


CERN DD/OC
Information Management: A Proposal
Tim Berners-Lee, CERN/DD
March 1989

Information Management: A Proposal

Abstract

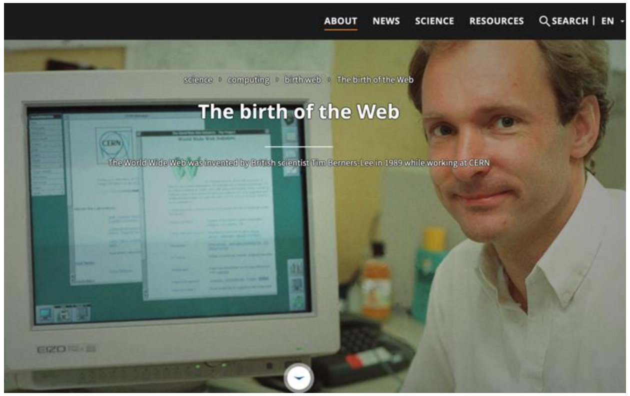
This proposal concerns the management of general information about accelerators and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a solution based on a distributed hypertext sytstem.

Keywords: Hypertext, Computer conferencing, Document retrieval, Information management, Project control

Tim Berners-Lee wrote the first proposal for the World Wide Web in March 1989 and his <u>second proposal in May 1990</u>. Together with Belgian systems engineer Robert Cailliau, this was formalised as <u>a management proposal</u> in November 1990.

The document described a "hypertext project" called "WorldWideWeb" in which a "web" of "hypertext documents" could be viewed by “browsers”.

By the end of 1990, Tim Berners-Lee had the first Web server and browser up and running at CERN, demonstrating his ideas.

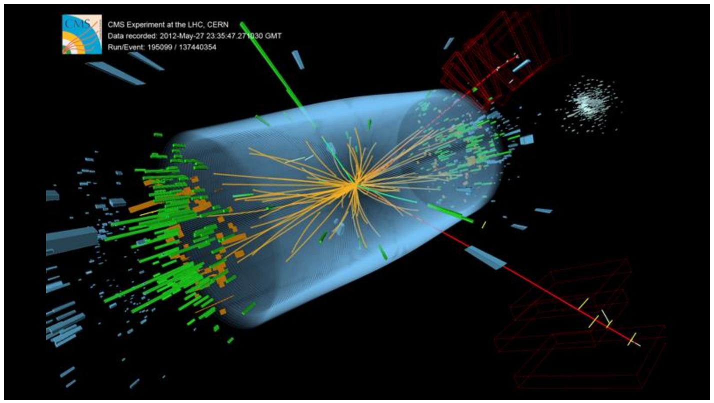


He developed the code for his Web server on a NeXT computer. To prevent it being accidentally switched off, the computer had a hand-written label in red ink: "This machine is a server. DO NOT POWER IT DOWN!!"

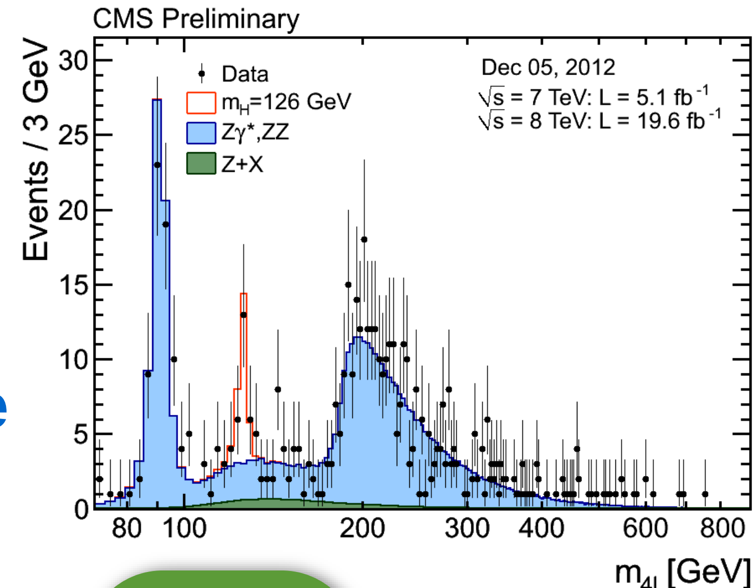
Experimental Particle Physics - the Journey

Particle Collisions

Higgs Boson Discover



Large Scale Computing



Device

Simulation

RAW Data

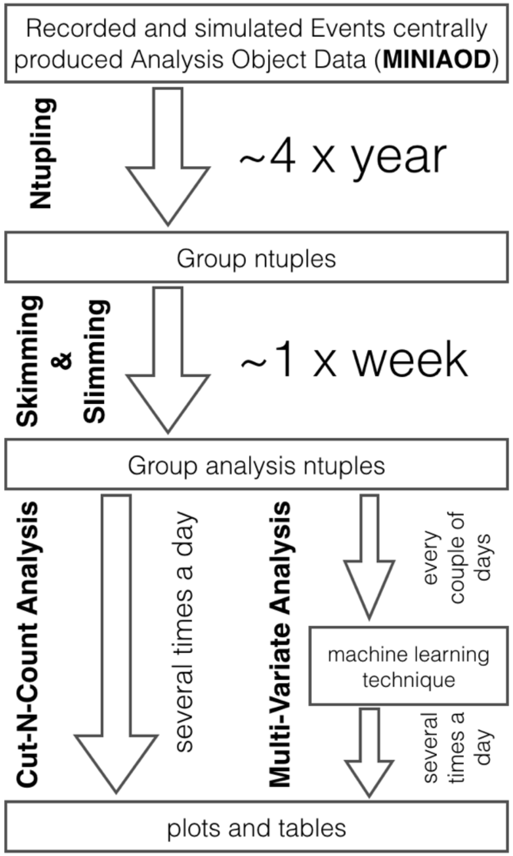
Algorithms to reconstruct data

RECO Data

Analysis software

Plots

Data Analysis: a multi-step process



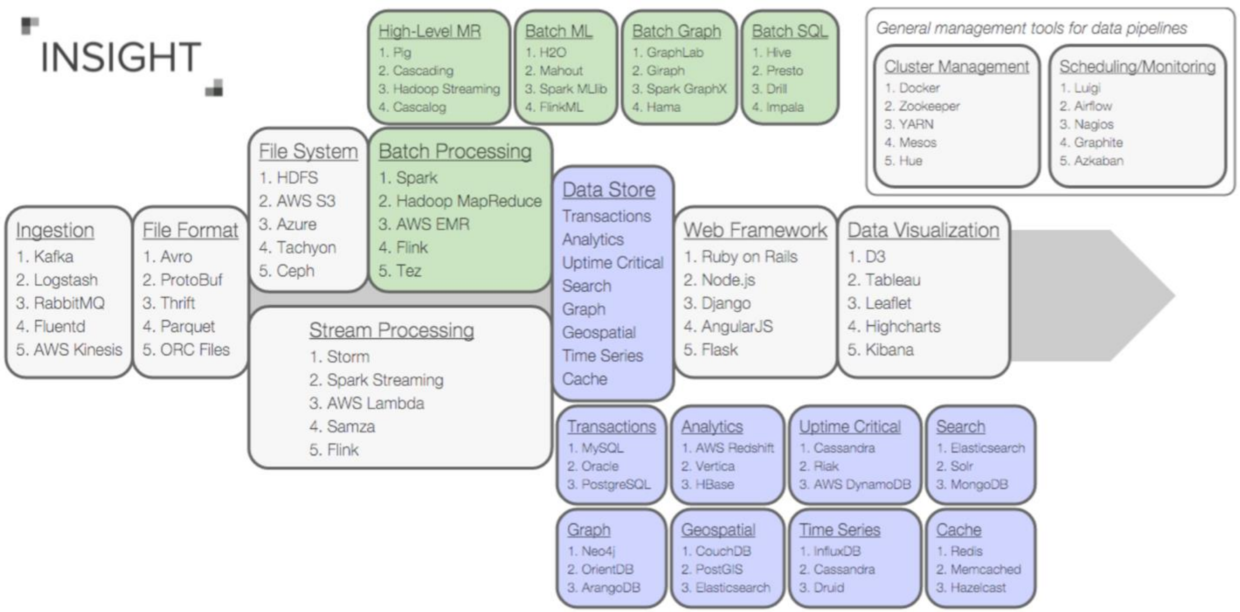
- Minimize **Time to Insight**
  - Analysis is a conversation with data - Interactivity is key
- Many different physics topics concurrently under investigation
  - Different slices of data are relevant for each analysis
- Programmatically same analysis steps
  - Skimming (dropping events in a disk-to-disk copy)
  - Slimming (dropping branches in a disk-to-disk copy)
  - Filtering (selectively reading events into memory)
  - Pruning (selectively reading branches into memory)

Big Data for HEP

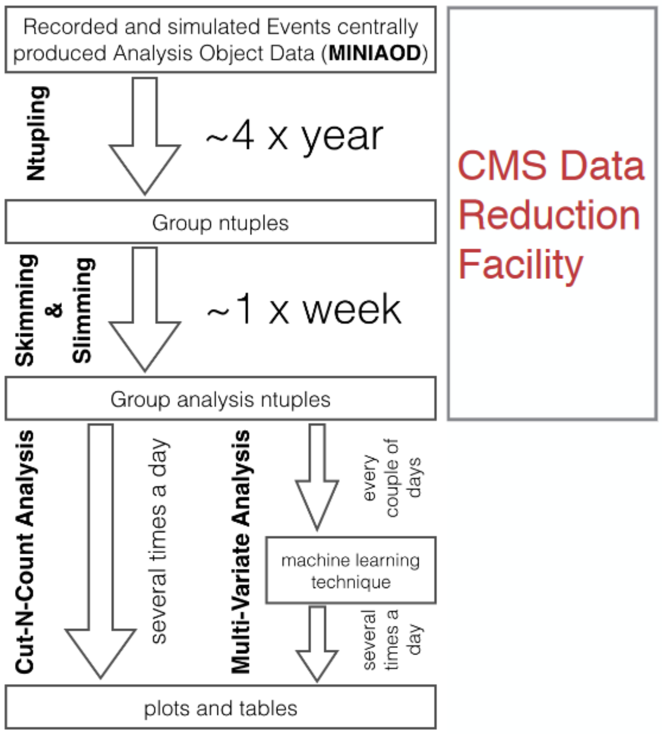
- New toolkits and systems collectively called "Big Data" technologies have emerged to support the analysis of PB and EB datasets in industry.
- ''Big data is a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications''

Our goals in applying these technologies to the HEP:

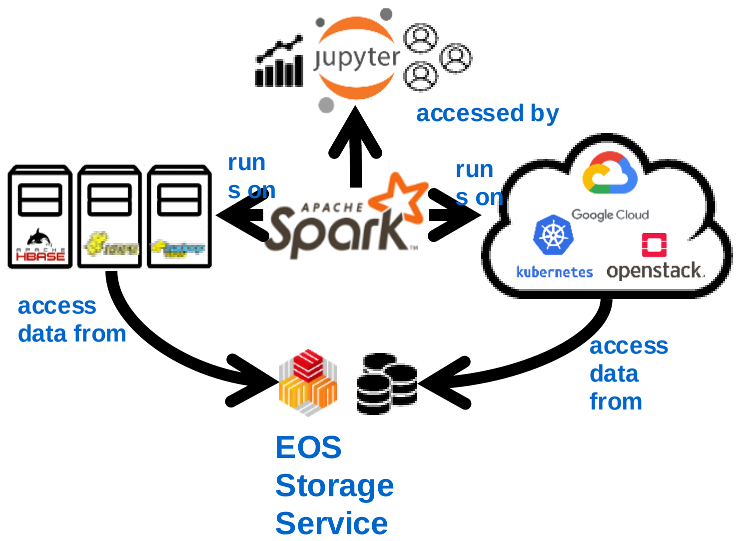
- **Reduce Time to Insight**
- **Educate our graduate students and post docs to use industry-based technologies**
- **Improves chances on the job market outside academia**
- **Increases the attractiveness of our field**
- **Be part of an even larger community**



Data Reduction and Analysis Facility



- CERN openlab / Intel project /Recas (Bari)
- Apache Spark is a unified analytics engine for large-scale data processing with built-in modules for
  - SQL, streaming, machine learning, and graph processing.
- Spark can run on Apache Hadoop, Apache Mesos, Kubernetes, on its own, in the cloud and for diverse data sources.



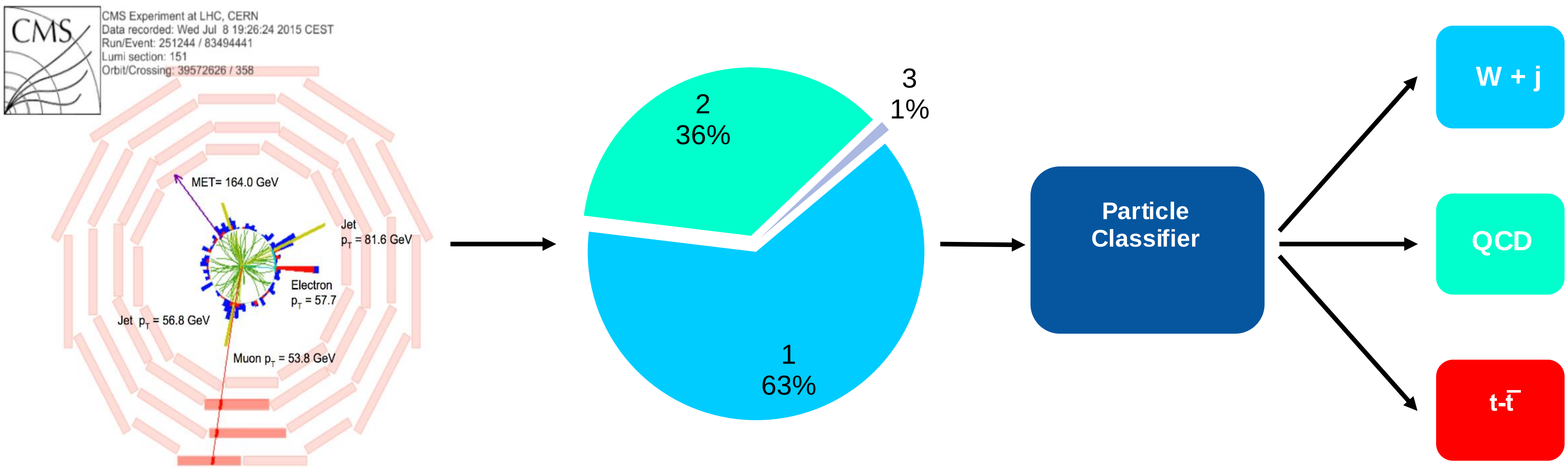
- **Goal: Demonstrate reduction capabilities producing analysis ntuples using Apache Spark**

Analytics Platform at CERN

- **Demonstrator's goal: data reduction of 1 PB input data in 5 hours**

Deep Learning Pipeline for Physics Data

- **R&D to improve the quality of filtering systems**
  - **Develop a "Deep Learning classifier" to be used by the filtering system**
  - **Goal: Reduce false positives → do not store nor process uninteresting events**
  - **"Topology classification with deep learning to improve real-time event selection at the LHC", Nguyen et al. Comput.Softw.Big Sci. 3 (2019) no.1, 12**



N. De Filippis

Engineering effort to enable Effective ML

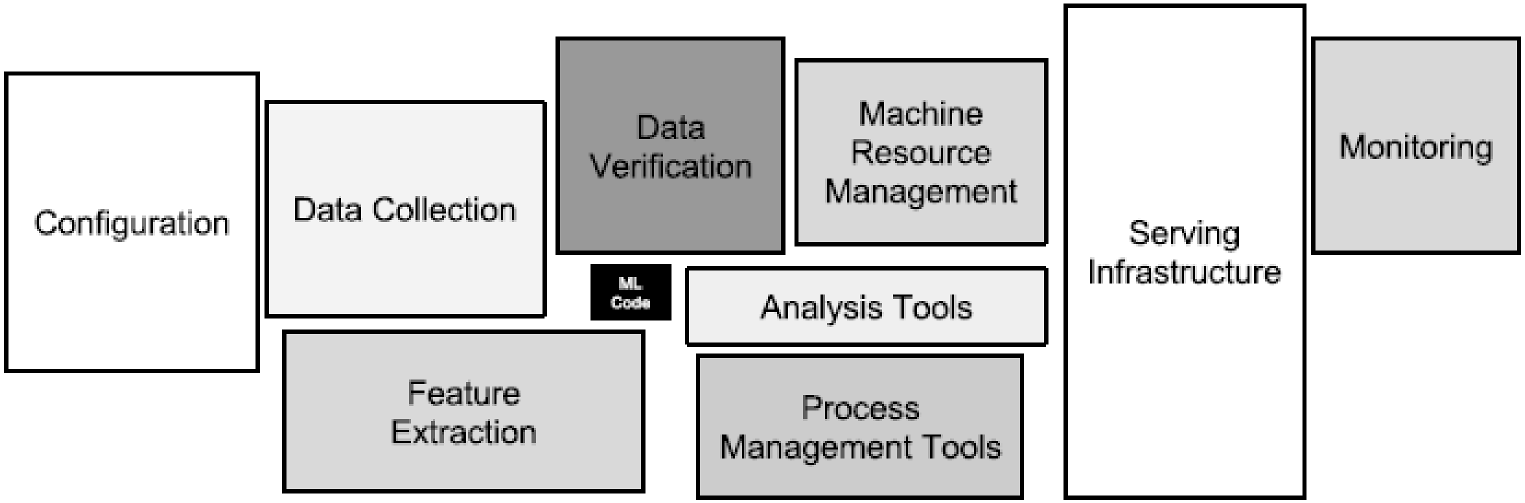


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

N. De Filippis

Deep Learning Pipeline for Physics Data

Data Ingestion → **Feature Preparation** → **Model Development** → **Training**

Data Ingestion

Read physics data and feature engineering

Feature Preparation

Prepare input for Deep Learning network

Model Development

1. Specify model topology
2. Tune model topology on small dataset

Training

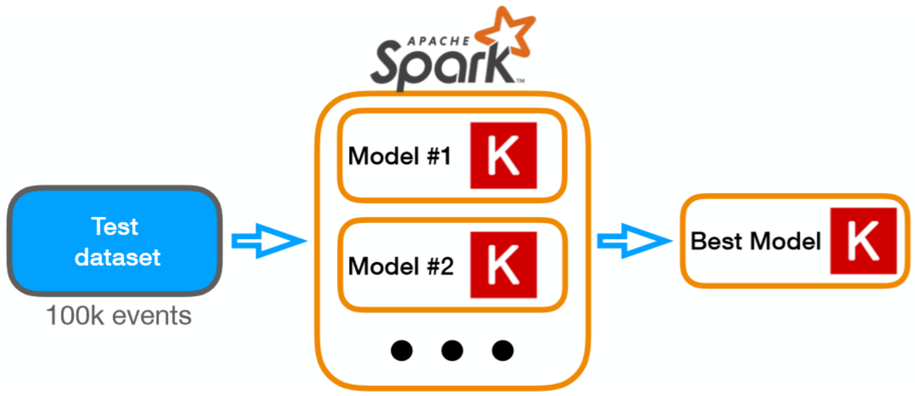
Train the best model

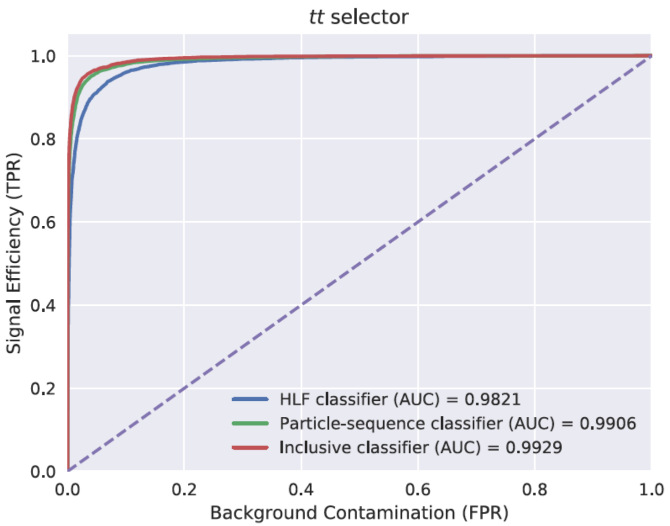
Built with Apache Spark + Analytics Zoo + Python Notebooks

N. De Filippis

Hyper-Parameter Tuning

- Once the network topology is chosen, hyper-parameter tuning is done with scikit-learn + Keras and parallelized with Spark
- the Area Under the ROC curve (AUC), as the performance metric to compare different classifiers
- the feed-forward DNN tuning done by changing the number of layers and units per layer, the activation function, the optimizer, etc.





N. De Filippis

Machine Learning with Spark and Keras

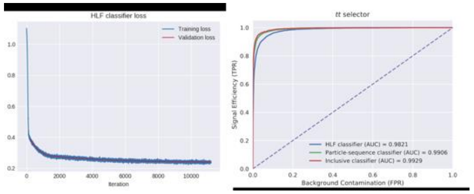
Data and models from Researchers

Input: labeled data and DL models

Feature engineering at scale

Hyperparameter optimization (Random/Grid search)

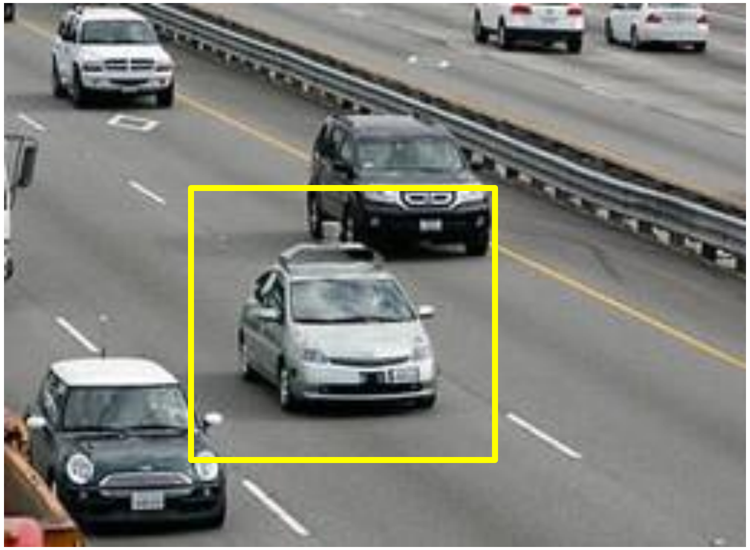
Distributed model training



Output: particle selector model

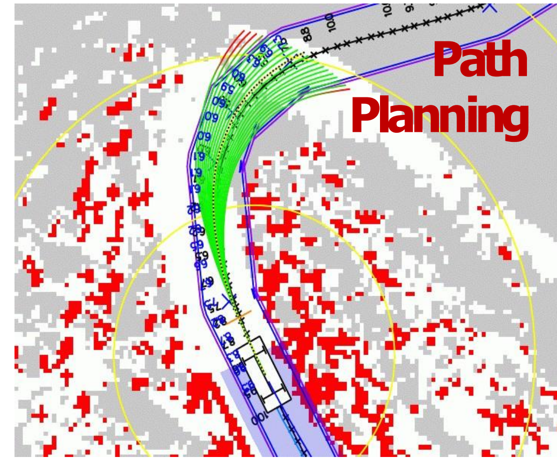
State of the Art Applications of Machine Learning for daily life

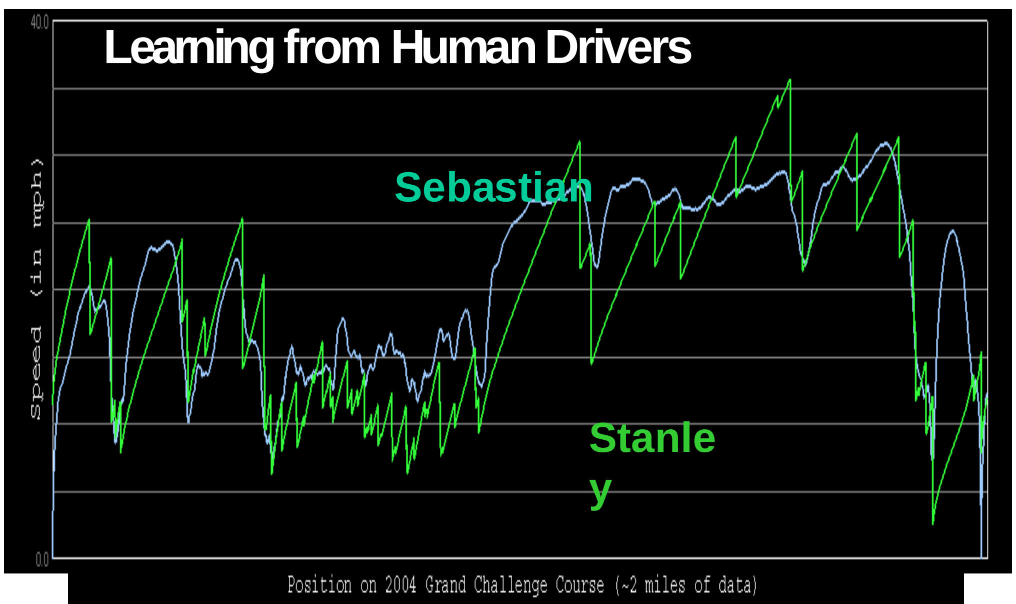
Autonomous cars



- Nevada made it legal for autonomous cars to drive on roads in June 2011
- As of 2013, four states (Nevada, Florida, California, and Michigan) have legalized autonomous cars

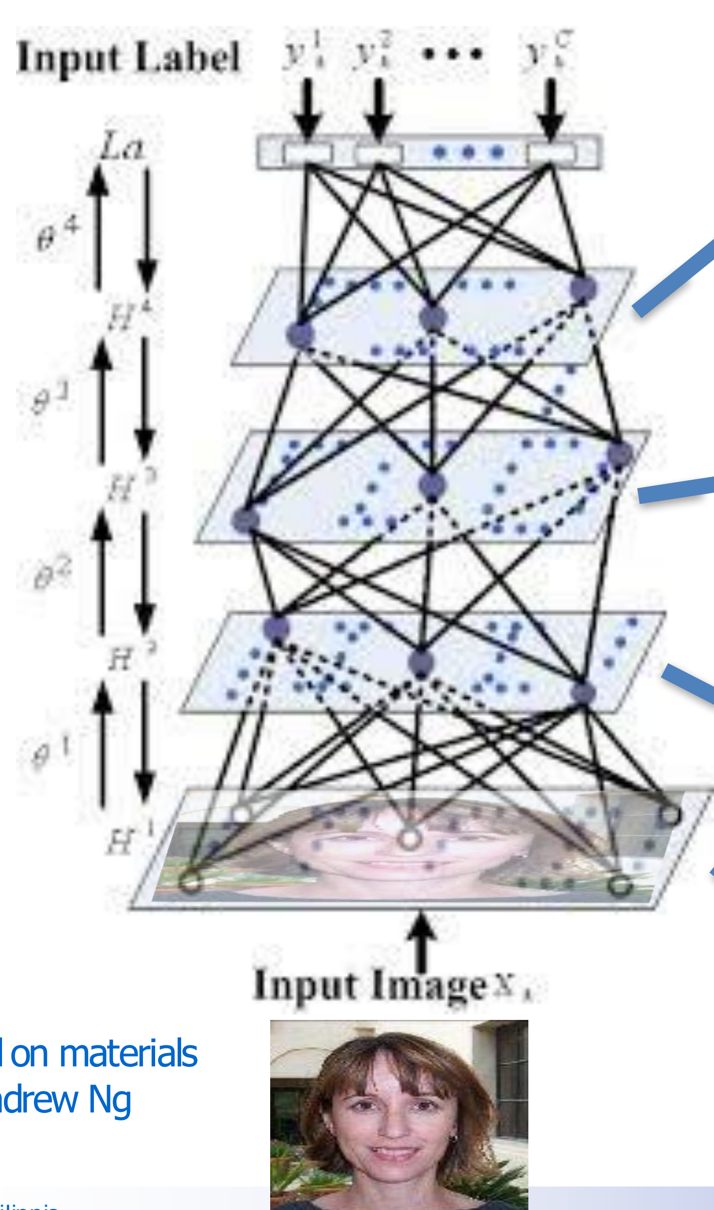
Autonomous car technology

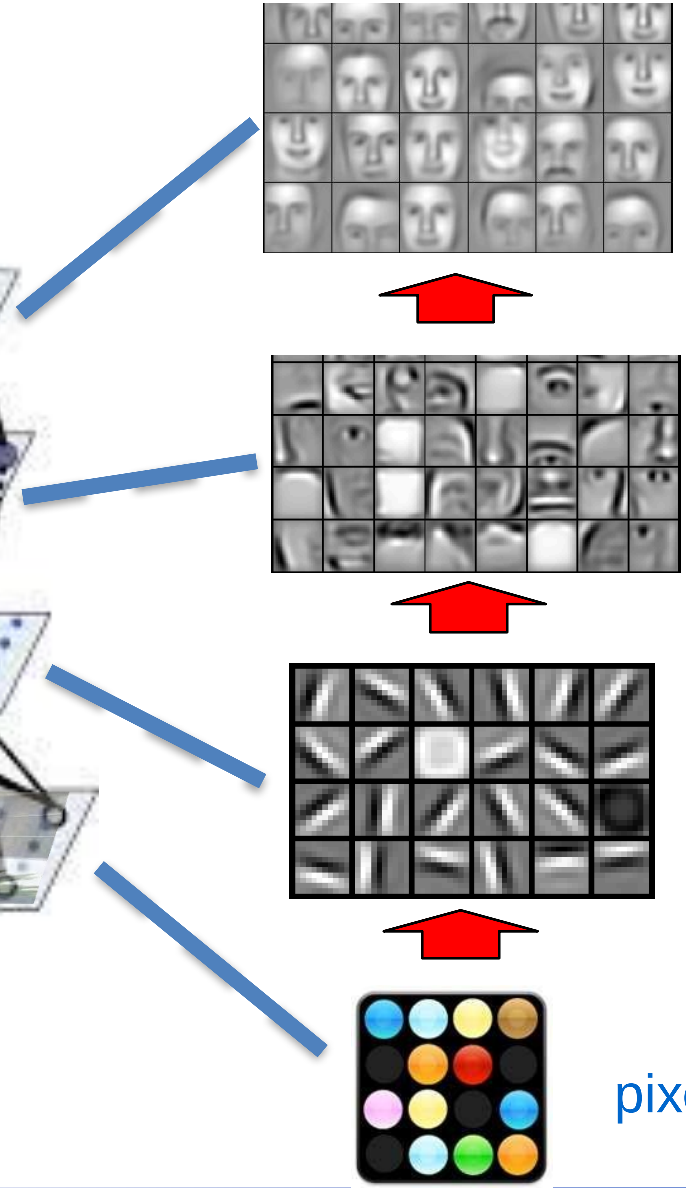




Images and movies taken from Sebastian Thrun's multimedia website

Deep Belief Net on Face Images





object models

object parts (combination of edges)

edges

pixels

Based on materials by Andrew Ng

Scene Labeling via Deep Learning

[Farabet et al. ICML 2012, PAMI 2013]

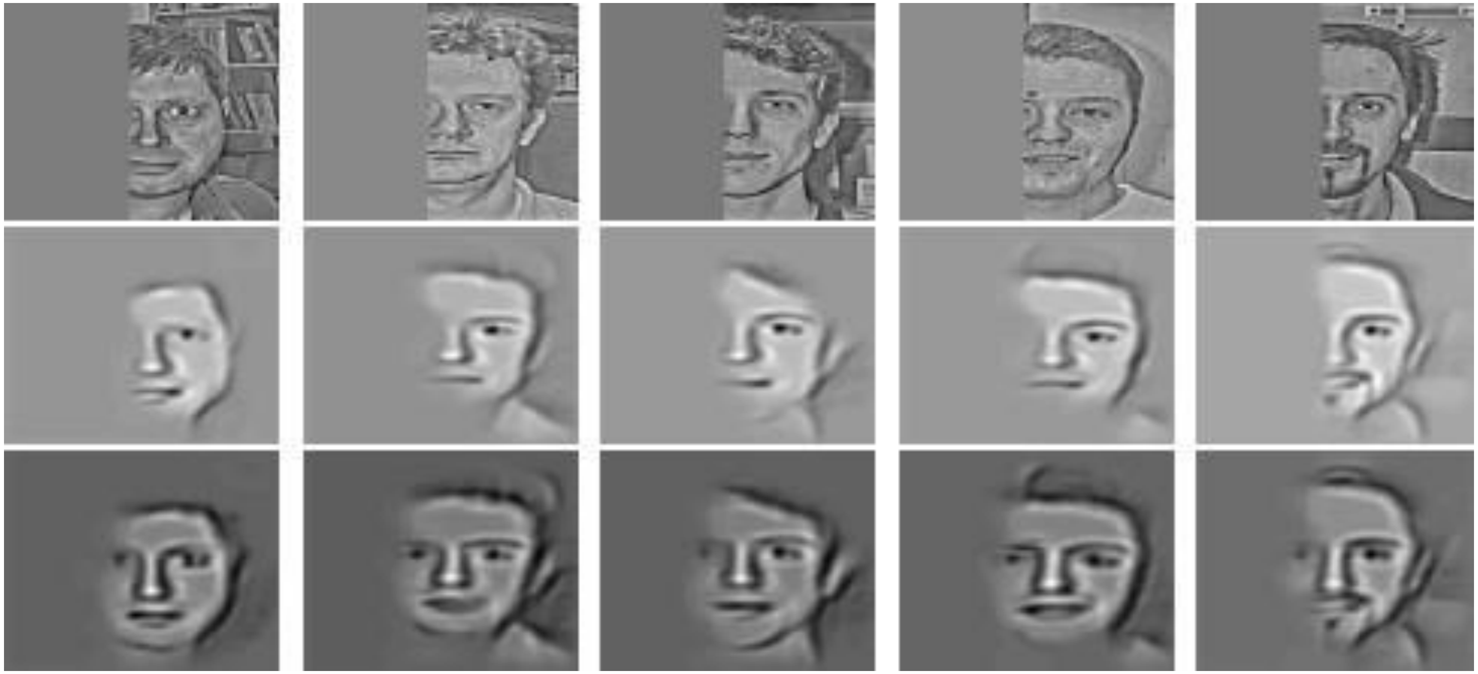
Inference from Deep Learned models

Generating posterior samples from faces by "filling in" experiments (cf. Lee and Mumford, 2003). Combine bottom-up and top-down inference.

Input images

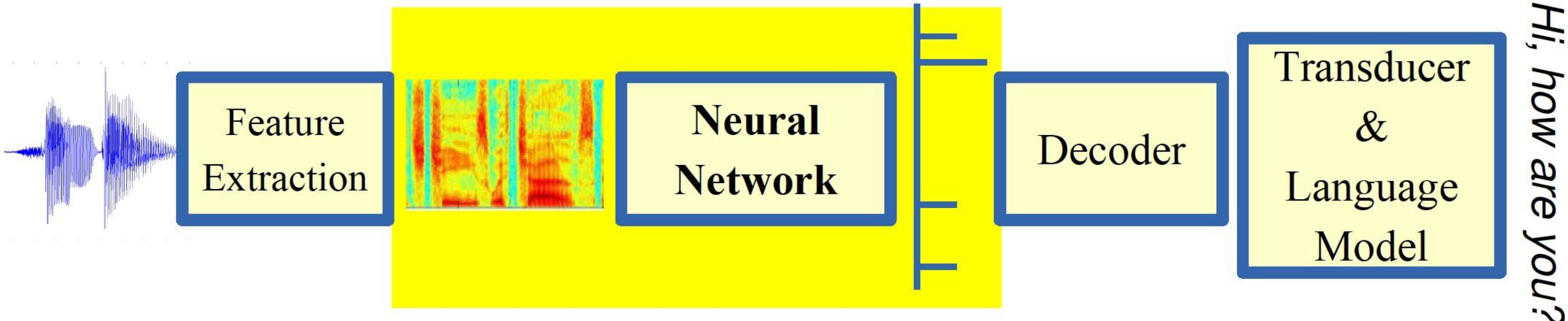
Samples from feedforward Inference (control)

Samples from Full posterior inference

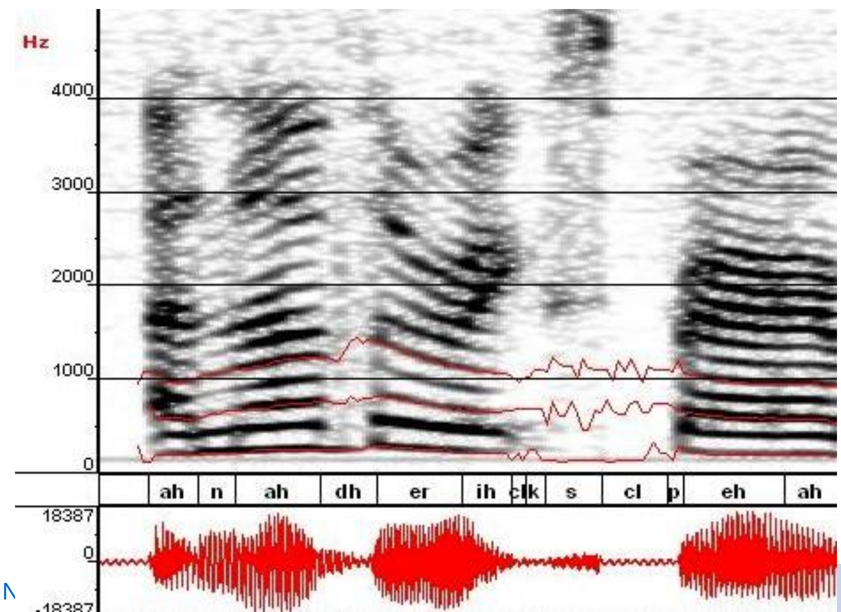


Machine Learning in Automatic Speech Recognition

A Typical Speech Recognition System



ML used to predict of phone states from the sound spectrogram



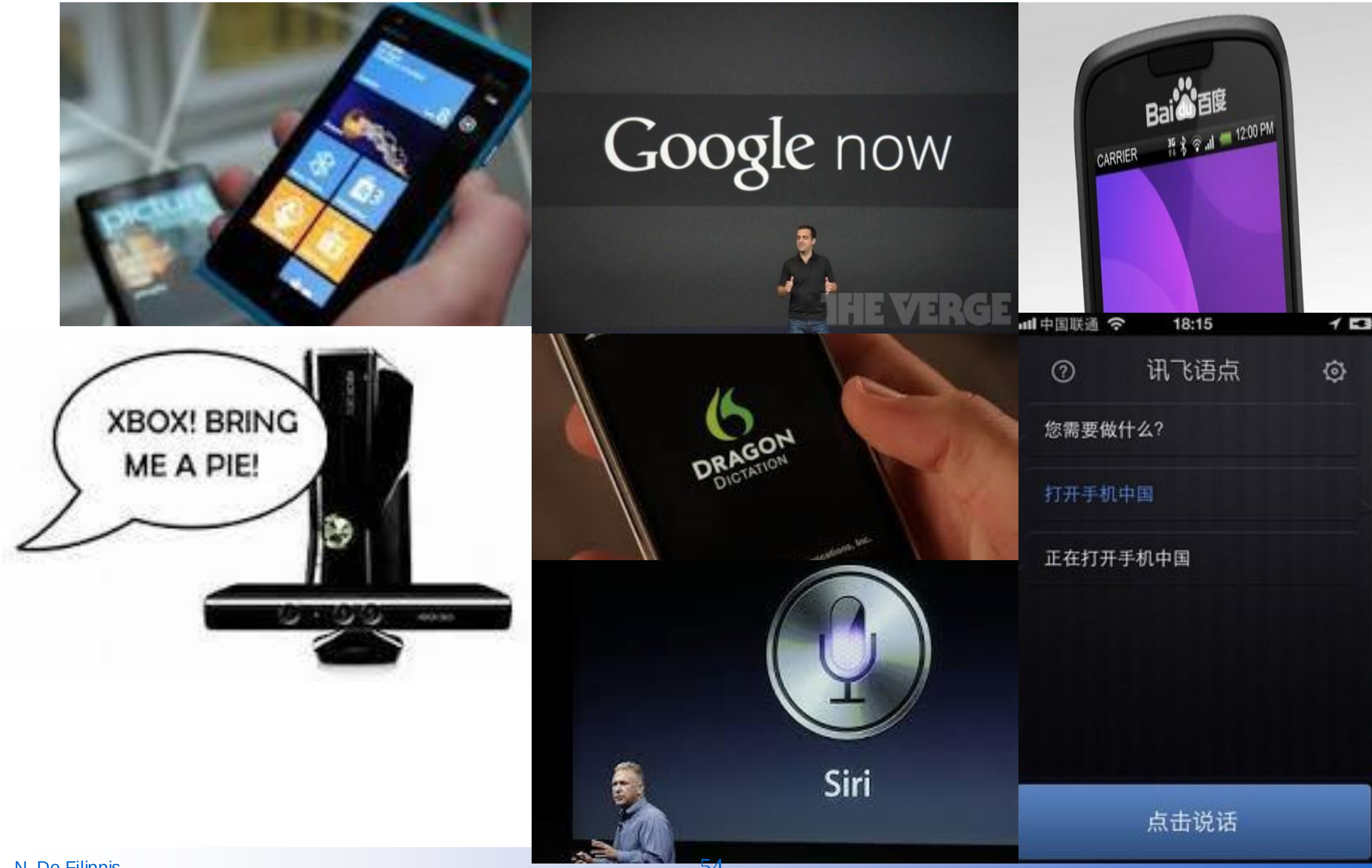
Deep learning has state-of-the-art results

# Hidden Layers	1	2	4	8	10	12
Word Error Rate %	16.0	12.8	11.4	10.9	11.0	11.1

Baseline GMM performance = 15.4%

[Zeiler et al. "On rectified linear units for speech recognition" ICASSP 2013]

Impact of Deep Learning in Speech Technology



Conclusions

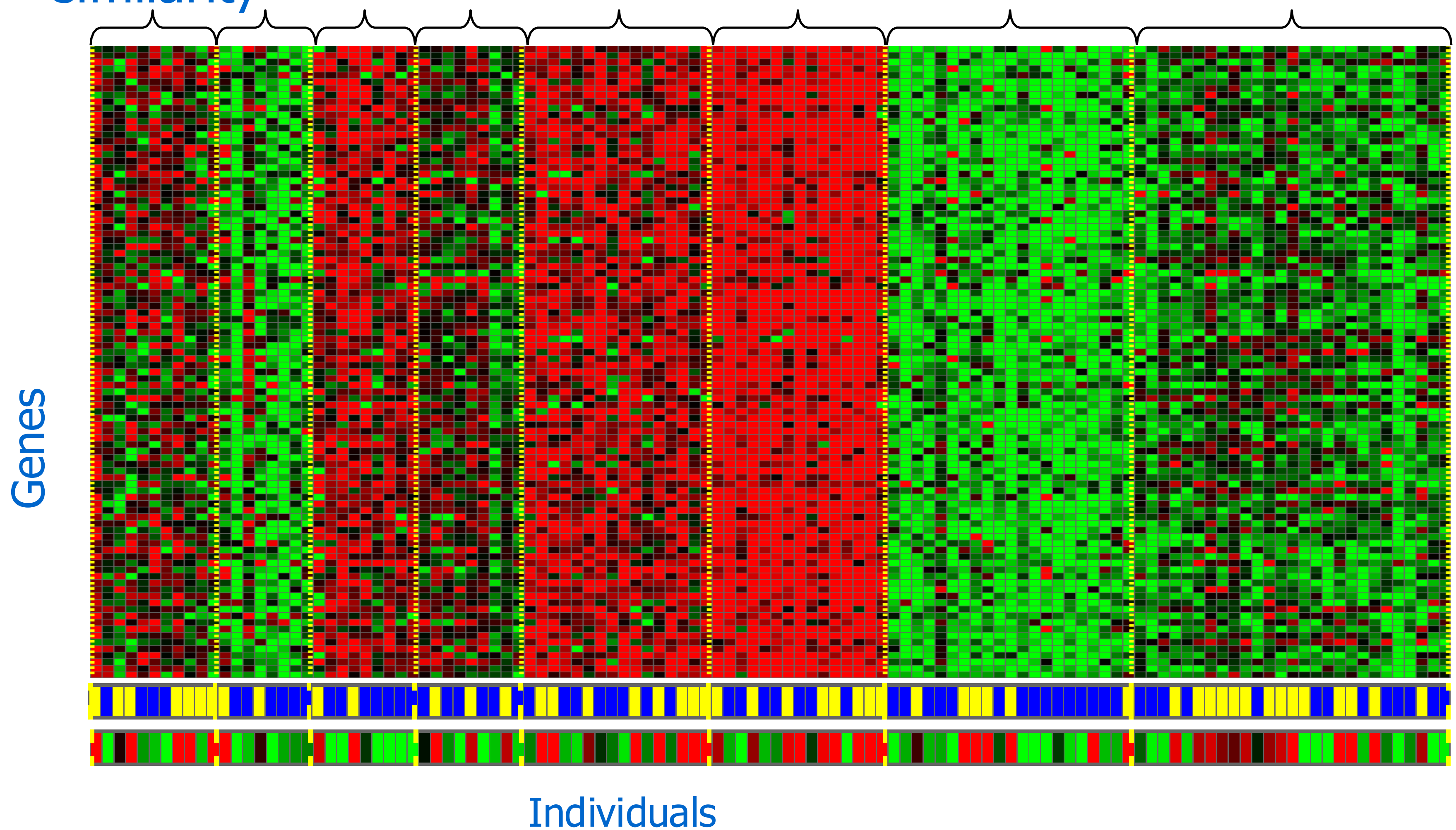
- Machine learning are part of our daily life and evolve rapidly for mutiple purposes and different complex problems.
- Wide variety of machine learning techniques available for collider classification, regression, and fast simulation tasks
- Feature-based classifiers widely used in HEP experiments and under study for future colliders
- Deep learning approach with low-level inputs has been shown to provide better performance for some problems
- Many different applications available on the market

Enjoy the benefit of ML in your daily life

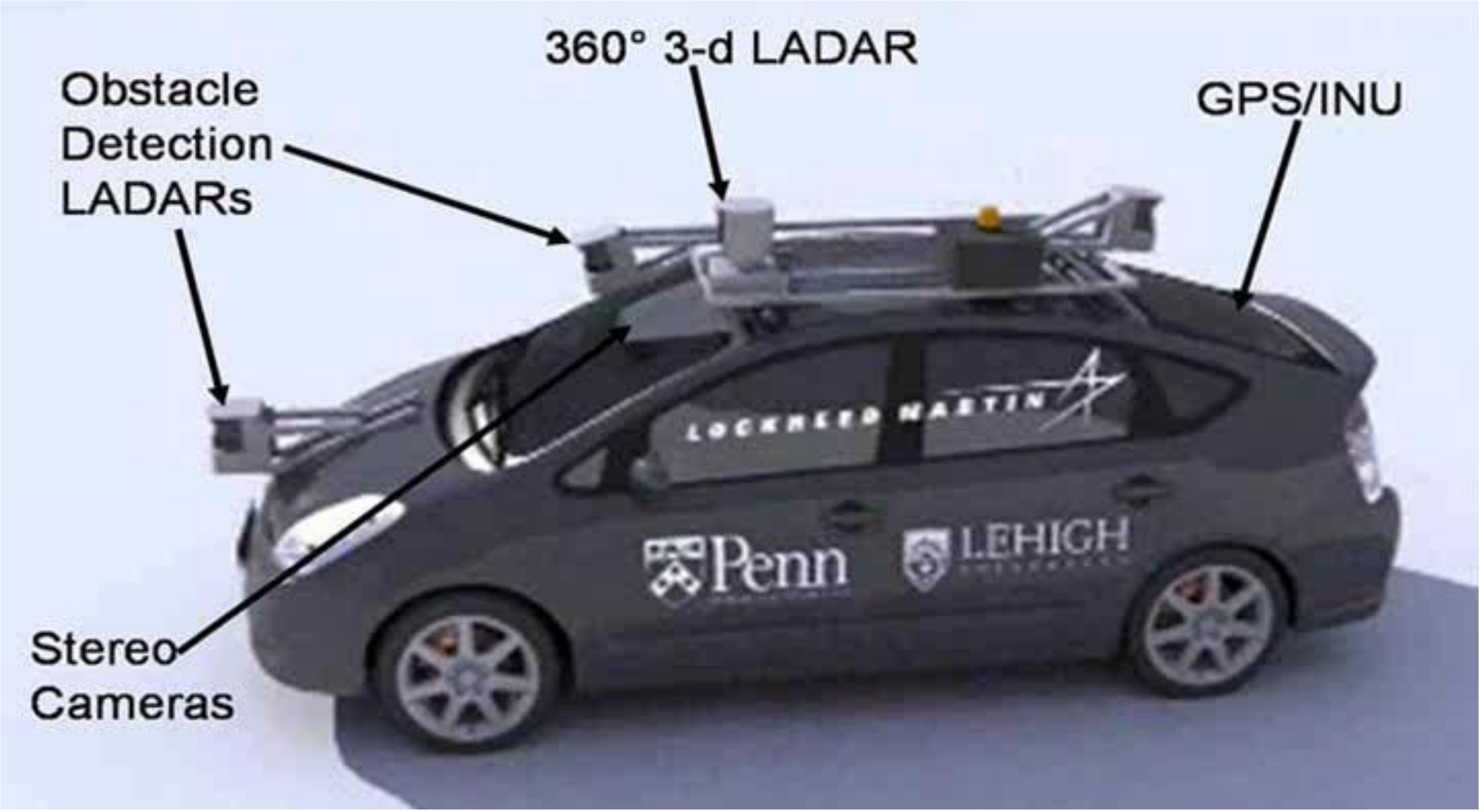
Backup

Unsupervised Learning

Genomics application: group individuals by genetic similarity



Autonomous car sensors



History of machine learning (1)

- 1950s
  - Samuel's checker player
  - Selfridge's Pandemonium
- 1960s:
  - Neural networks: Perceptron
  - Pattern recognition
  - Learning in the limit theory
  - Minsky and Papert prove limitations of Perceptron
- 1970s:
  - Symbolic concept induction
  - Winston's arch learner
  - Expert systems and the knowledge acquisition bottleneck
  - Quinlan's ID3
  - Michalski's AQ and soybean diagnosis
  - Scientific discovery with BACON
  - Mathematical discovery with AM

History of machine learning (2)

- 1980s:
  - Advanced decision tree and rule learning
  - Explanation-based Learning (EBL)
  - Learning and planning and problem solving
  - Utility problem
  - Analogy
  - Cognitive architectures
  - Resurgence of neural networks (connectionism, backpropagation)
  - Valiant's PAC Learning Theory
  - Focus on experimental methodology
- 1990s
  - Data mining
  - Adaptive software agents and web applications
  - Text learning
  - Reinforcement learning (RL)
  - Inductive Logic Programming (ILP)
  - Ensembles: Bagging, Boosting, and Stacking
  - Bayes Net learning

History of machine learning (3)

- 2000s
  - Support vector machines & kernel methods
  - Graphical models
  - Statistical relational learning
  - Transfer learning
  - Sequence labeling
  - Collective classification and structured outputs
  - Computer Systems Applications (Compilers, Debugging, Graphics, Security)
  - E-mail management
  - Personalized assistants that learn
  - Learning in robotics and vision
- 2010s
  - Deep learning systems
  - Learning for big data
  - Bayesian methods
  - Multi-task & lifelong learning
  - Applications to vision, speech, social networks, learning to read, etc.
  - ???

EVALUATION METRICS

- The idea of building machine learning models works on a constructive feedback principle:
  - building a model, getting a feedback from metrics, making improvements and continuing until you achieve the desired accuracy
- An important aspect of evaluation metrics is their capability to discriminate among model results
- The real goal is creating and selecting a model which gives high accuracy on sample data:
  - It is crucial to check the accuracy of your model prior to computing predicted values.

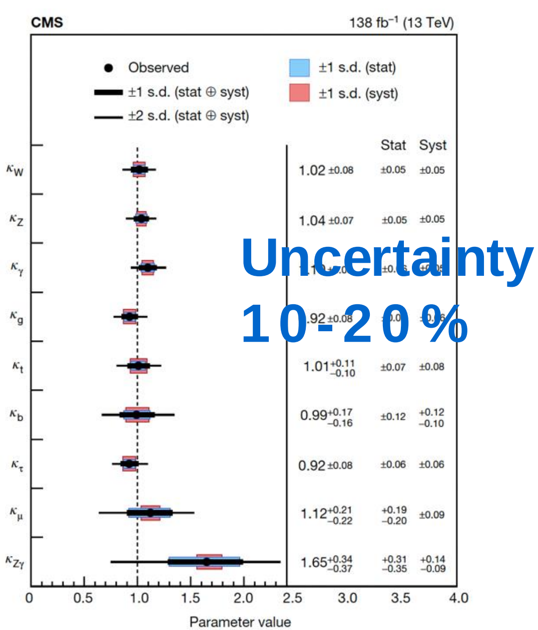
Evaluation

- **Accuracy**
- **Precision and recall**
- **Squared error**
- **Likelihood**
- **Posterior probability**
- **Cost / Utility**
- **Margin**
- **Entropy**
- **K-L divergence**
- **etc.**

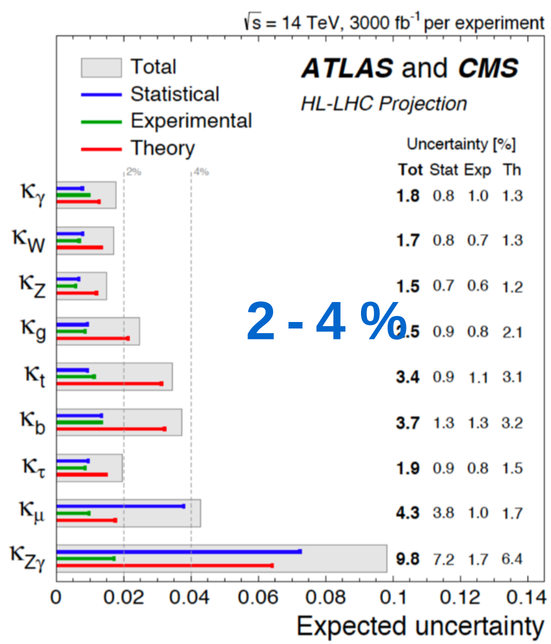
Landscape of the Higgs physics

So far many questions still open for Higgs physics:

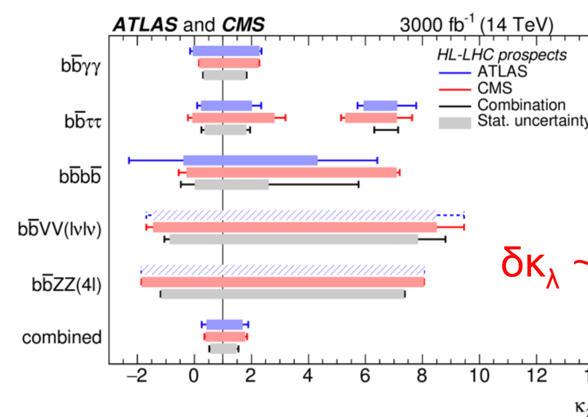
- ✓ How well the Higgs boson couplings to fermions, gauge bosons and to itself be probed at current, HL-LHC and future colliders?
- ✓ How do precision electroweak observables provide us information about the H properties and/or BSM physics?
- ✓ What progress is needed in theoretical developments in QCD and EWK to fully capitalize on the experimental data?
- ✓ What is the best path towards measuring the Higgs potential ?
- ✓ To what extent can we tell whether the Higgs is fundamental or composite?



Uncertainty 10-20%



2 - 4 %



$\delta\kappa_\lambda$ ~ 50%

- ➢ **Beyond HL-LHC:**
  - ✓ Couplings to fermions to %-level, to bosons to per-mil
  - ✓ self-coupling
  - ✓ Invisible decays, BSM Higgses

FCC long-term program

2020 ES for HEP:

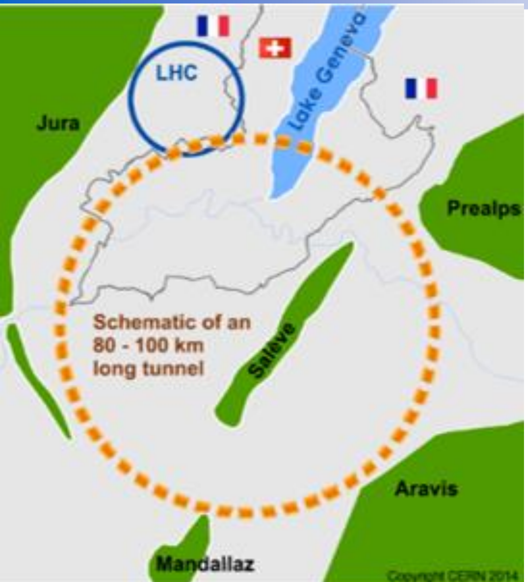
*"**An electron-positron Higgs factory is the highest priority next collider**. For the longer term, the European particle physics community has the ambition to operate a proton-proton collider at the highest achievable energy."*

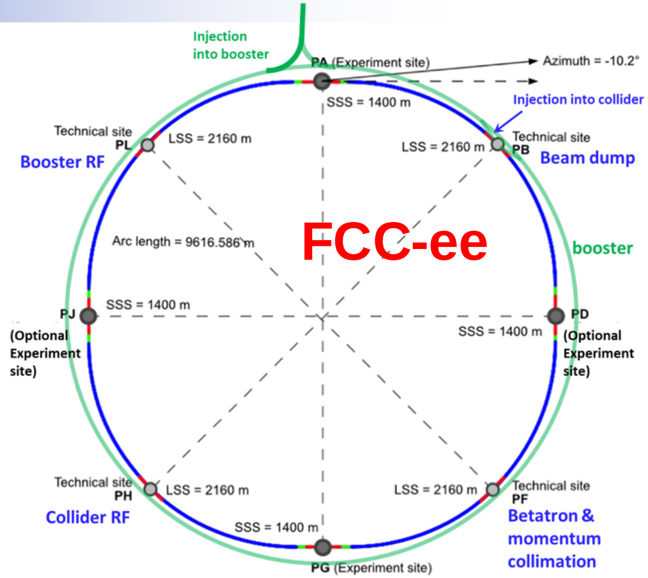
"Europe, together with its international partners, should investigate the technical and financial feasibility of a future hadron collider at CERN with a centre-of-mass energy of at least 100 TeV and with an electron-positron Higgs and electroweak factory as a possible first stage. Such a feasibility study of the colliders and related infrastructure should be established as a global endeavour and be completed on the timescale of the next Strategy update."

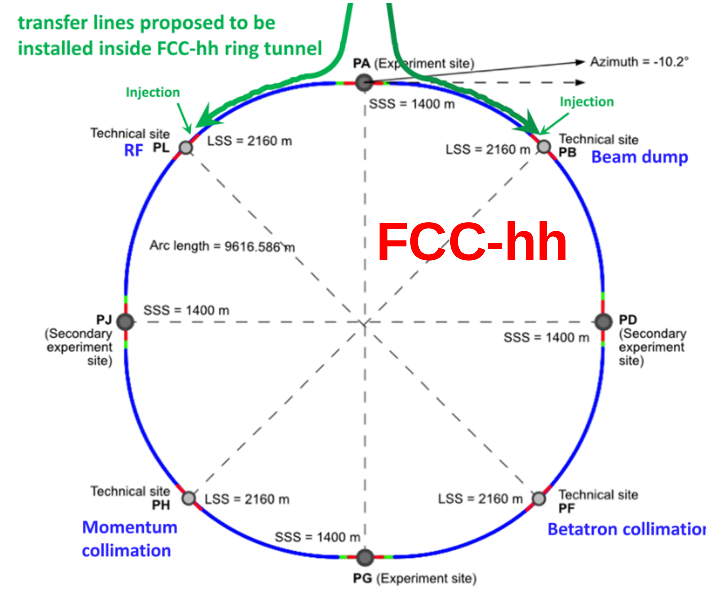
FCC@CERN: comprehensive program maximizing physics opportunities

- **stage 1: FCC-ee (Z, W, H, $t\bar{t}$) as Higgs factory, electroweak & top factory at highest luminosities**
- **stage 2: FCC-hh (~100 TeV) as natural continuation at energy frontier, pp & AA collisions; e-h option**
- highly synergetic and complementary programme boosting the physics reach of both colliders
- FCC integrated project allows the start of a new, major facility at CERN within a few years of the end of HL-LHC

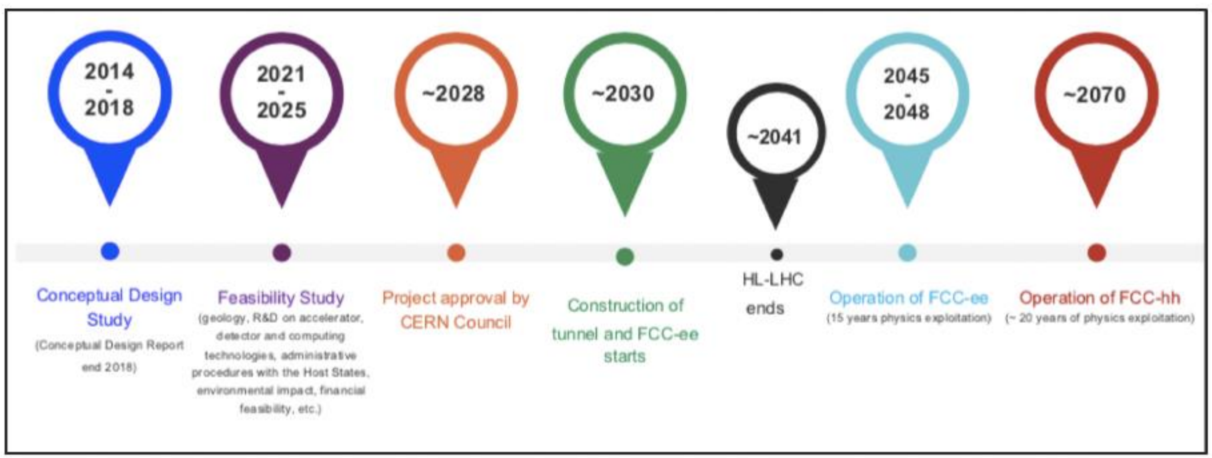
FCC long-term program

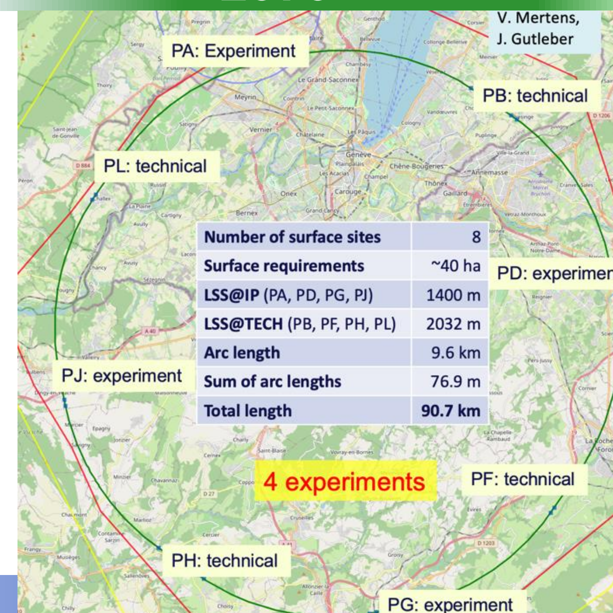






2020 - 2045 | **2045 - 2065** | **2070 -**

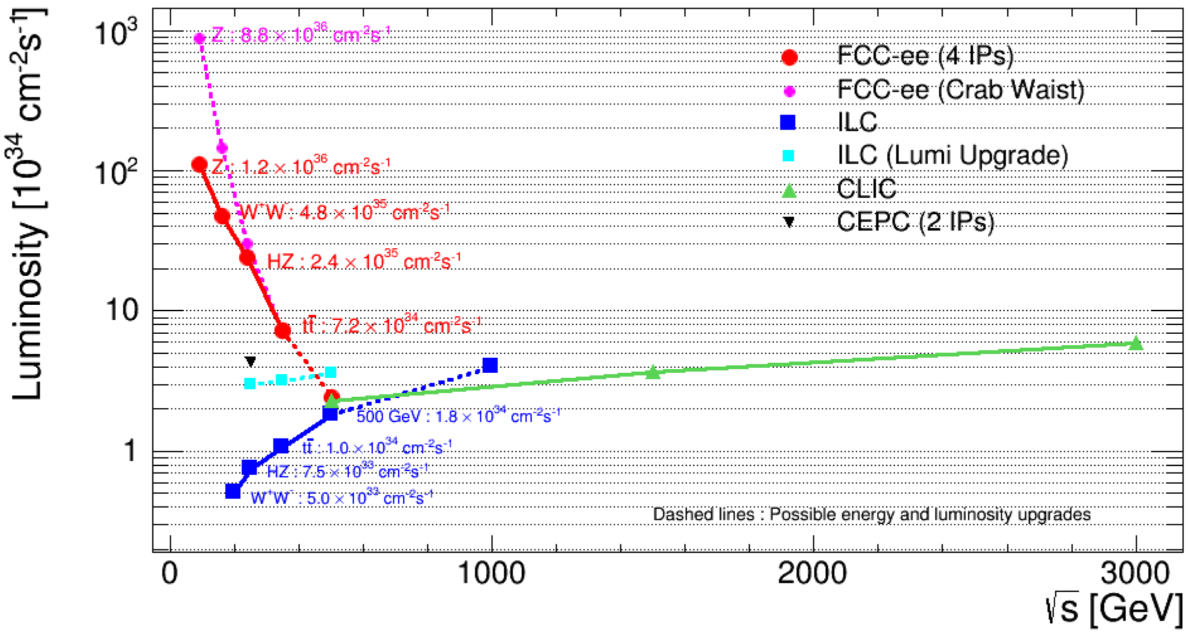




Number of surface sites	8
Surface requirements	~40 ha
LSS@IP (PA, PD, PG, PJ)	1400 m
LSS@TECH (PB, PF, PH, PL)	2032 m
Arc length	9.6 km
Sum of arc lengths	76.9 m
Total length	90.7 km

Final closest deliverable is a **Feasibility Study Report by March 2025**.

Machine luminosity for physics at e^+e^- colliders



- Higgs factory:
  - 10^6 $e^+e^- \to HZ$
- EW & Top factory:
  - 3×10^{12} $e^+e^- \to Z$
  - 10^8 $e+e^- \to W^+W^-$
  - 10^6 $e^+e^- \to tt$
- Flavor factory:
  - 5×10^{12} $e^+e^- \to bb, cc$
  - 10^{11} $e^+e^- \to \tau^+\tau^-$

~100 kHz of physics data at the Z pole

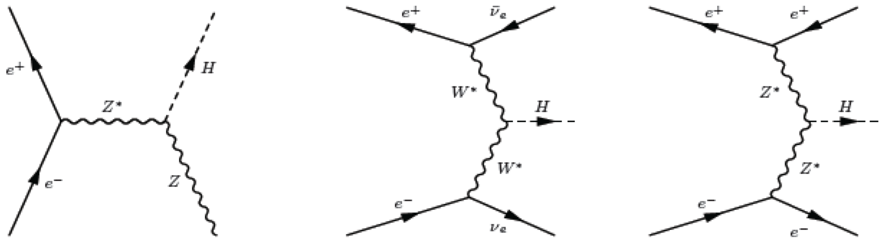
Phase	Run duration (years)	Center-of-mass Energies (GeV)	Integrated Luminosity (ab^{-1})	Event Statistics	
FCC-ee-Z	4	88-95 ± <100 KeV	150	3×10^{12} visible Z decays	LEP * 10^5
FCC-ee-W	2	158-162 <200 KeV	12	10^8 WW events	LEP * 2.10^3
FCC-ee-H	3	240 ± 2 MeV	5	10^6 ZH events	Never done
FCC-ee-tt	5	345-365 ± 5 MeV	1.5	10^6 $t\bar{t}$ events	Never done
s channel H	?	125 ± 2 MeV	10?	5000 events	Never done

Extracted from FCC CDR

$$\approx \frac{\Delta_{LEP,Stat}}{500}$$

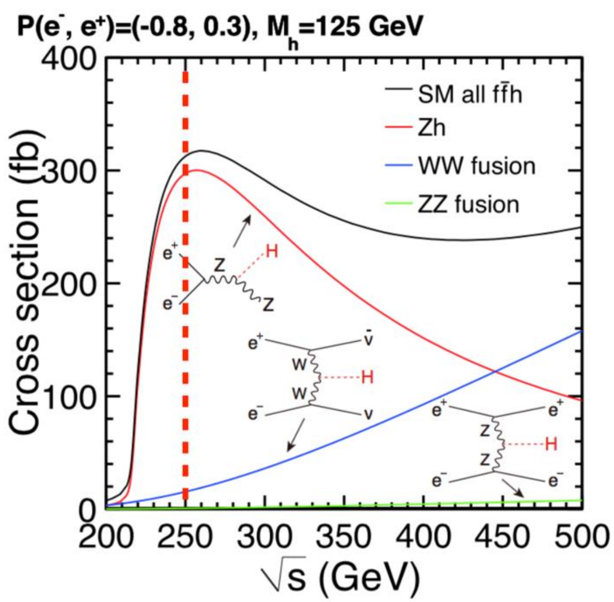
Higgs production at FCC-ee

Higgs-strahlung or $e^+e^- \to ZH$



Higgs production @ FCC-ee		
Threshold	ZH production	VBF production
240 GeV / 5 ab^{-1}	1e6	2.5e4
365 GeV / 1.5 ab^{-1}	2e5	5e4

VBF production: $e^+e^- \to \nu\nu H$ (WW fus.), $e^+e^- \to He^+e^-$ (ZZ fus.)



Process	Cross section	Events in 5 ab^{-1}
Higgs boson production, cross section in fb		
$e^+e^- \to ZH$	212	1.06×10^6
$e^+e^- \to \nu\bar{\nu}H$	6.72	3.36×10^4
$e^+e^- \to e^+e^-H$	0.63	3.15×10^3
Total	219	1.10×10^6
Background processes, cross section in pb		
$e^+e^- \to e^+e^-$ (Bhabha)	25.1	1.3×10^8
$e^+e^- \to q\bar{q}$	50.2	2.5×10^8
$e^+e^- \to \mu\mu$ (or $\tau\tau$)	4.40	2.2×10^7
$e^+e^- \to WW$	15.4	7.7×10^7
$e^+e^- \to ZZ$	1.03	5.2×10^6
$e^+e^- \to eeZ$	4.73	2.4×10^7
$e^+e^- \to e\nu W$	5.14	2.6×10^7