Machine Learning for HEP and more

Nicola De Filippis
Politecnico and INFN Bari

"Data Science Applications in Physics" Balkan School in Tirana 2024

Tirana, January 22-26, 2024

Introduction to Machine Learning

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

- Herbert Simon

Definition by Tom Mitchell (1998):

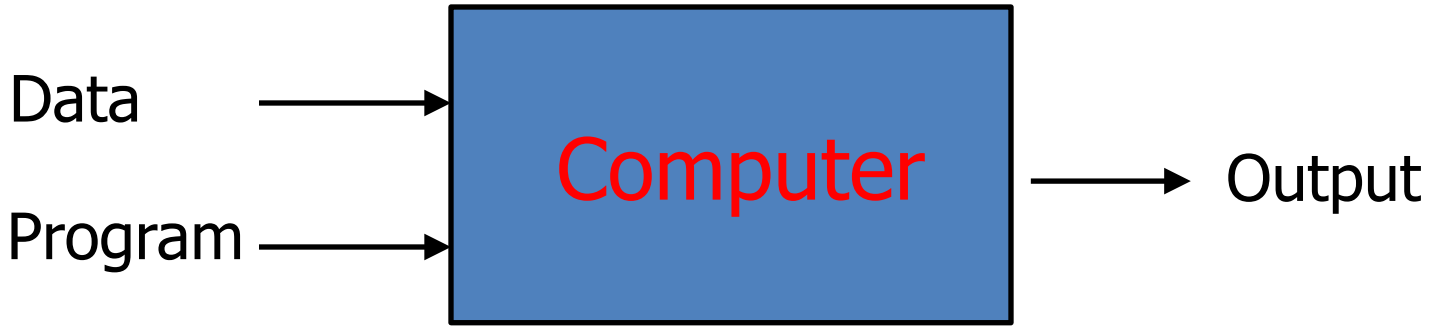
Machine Learning is the study of algorithms that

- improve their performance P
- at some task T
- with experience E.

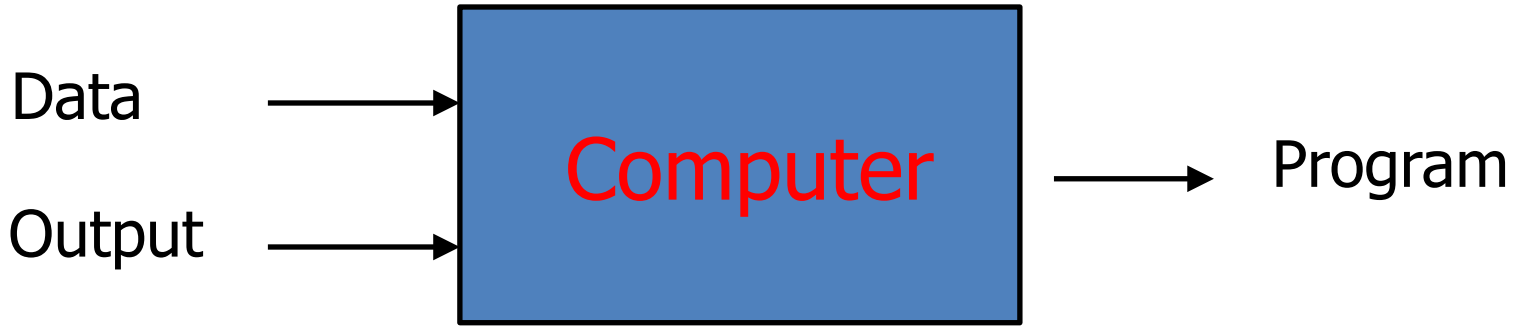
A well-defined learning task is given by <P, T, E>.

Comparison of different approaches

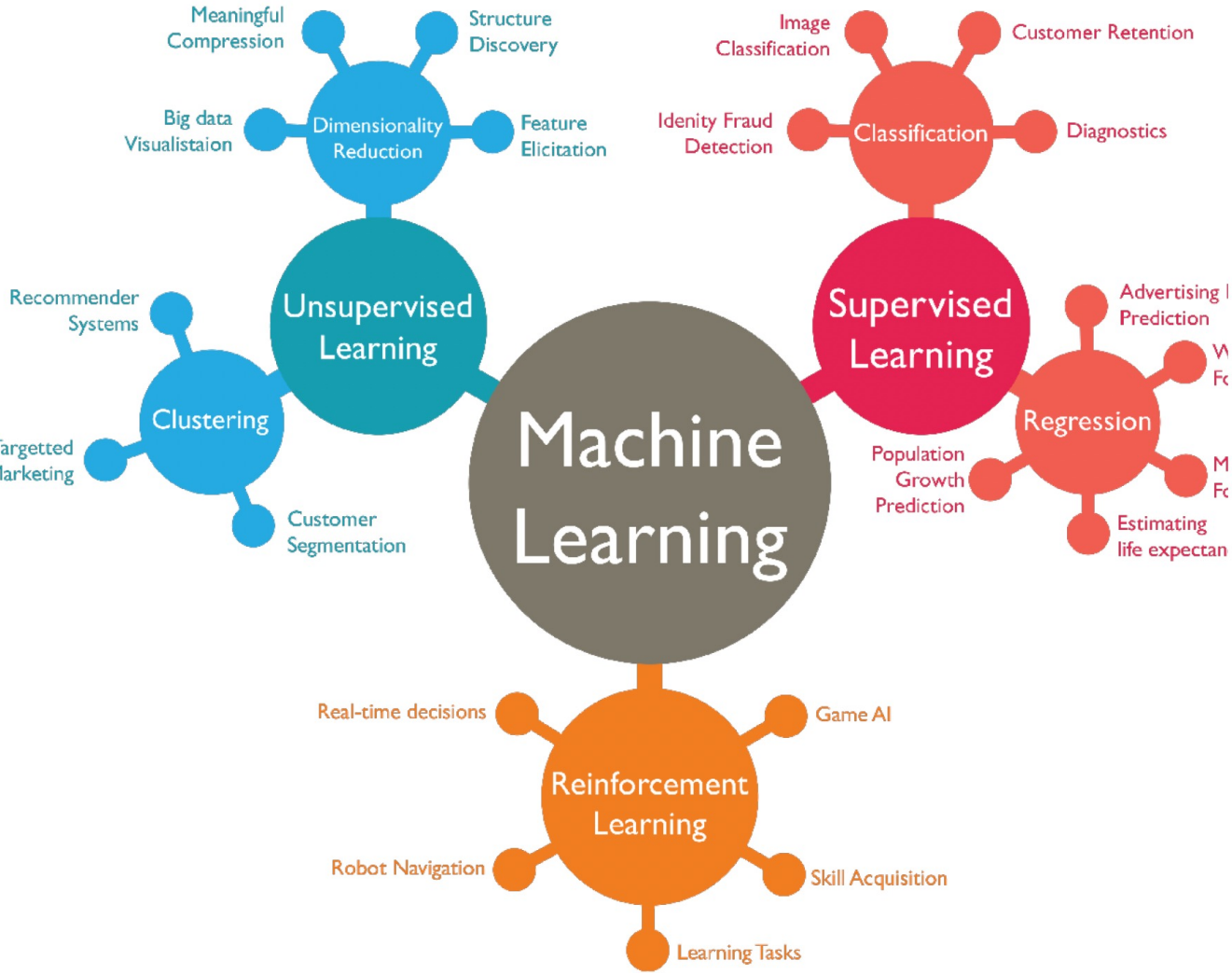
Traditional programming



Machine learning



Machine Learning world



WHAT IS MACHINE LEARNING?

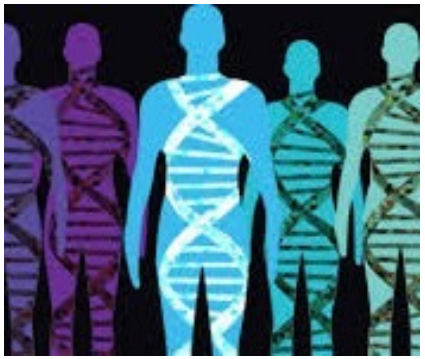
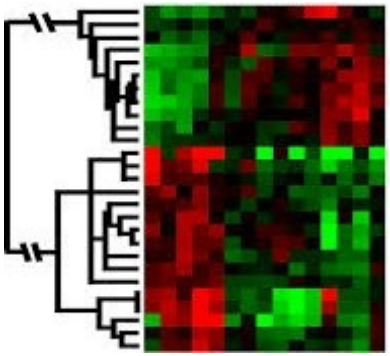
Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so.

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

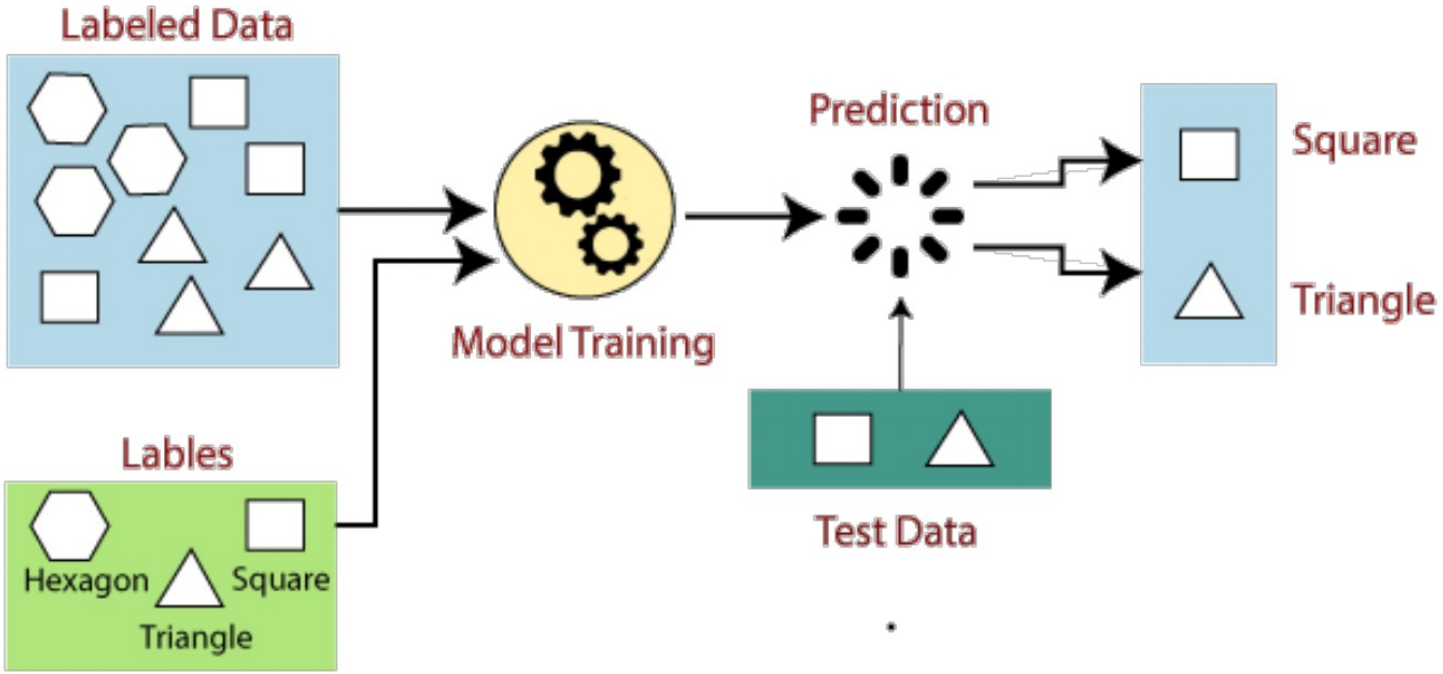
- Web search
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Debugging software
- Fundamental Science → HEP

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.



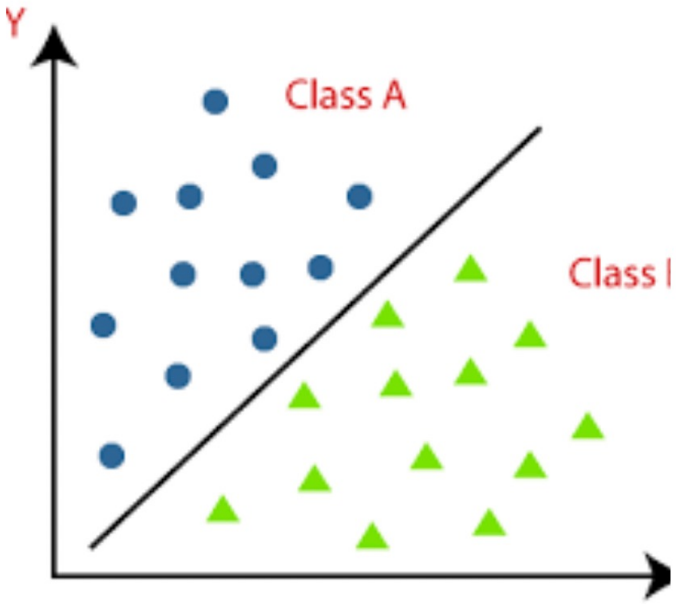
Supervised Learning: Regression

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

SCORE PREDICTIONS

Supervised Learning: 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*

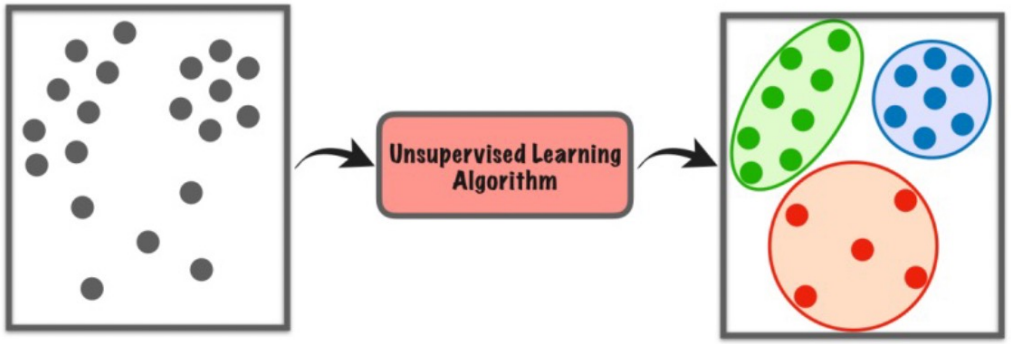


Unsupervised Learning

*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

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

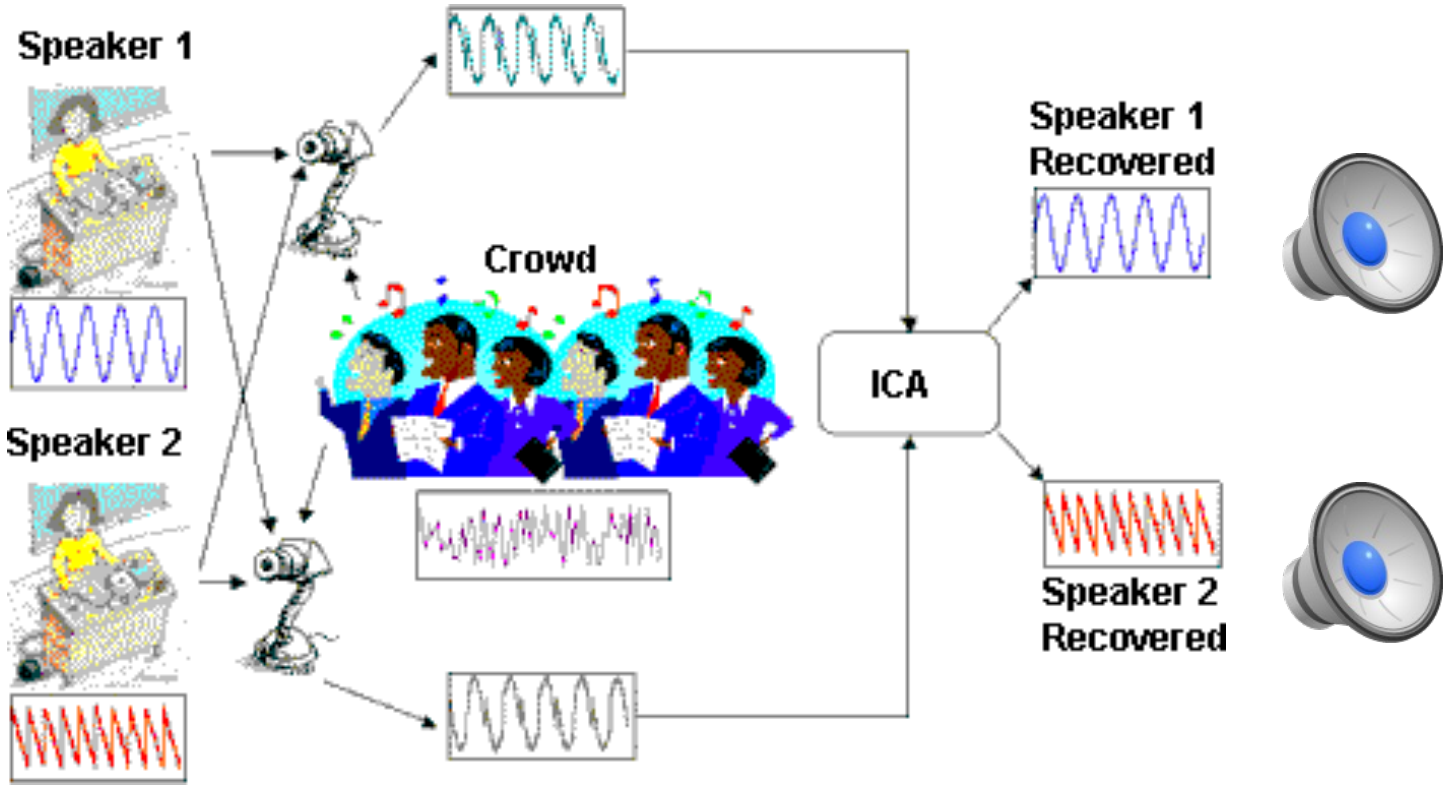
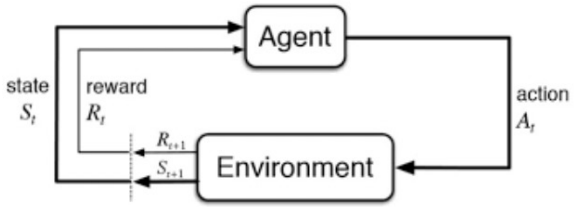


Image credit: statsoft.com Audio from
http://www.ism.ac.jp/~shiro/research/blindsep.html

Reinforcement Learning



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

→

PACMAN GAME

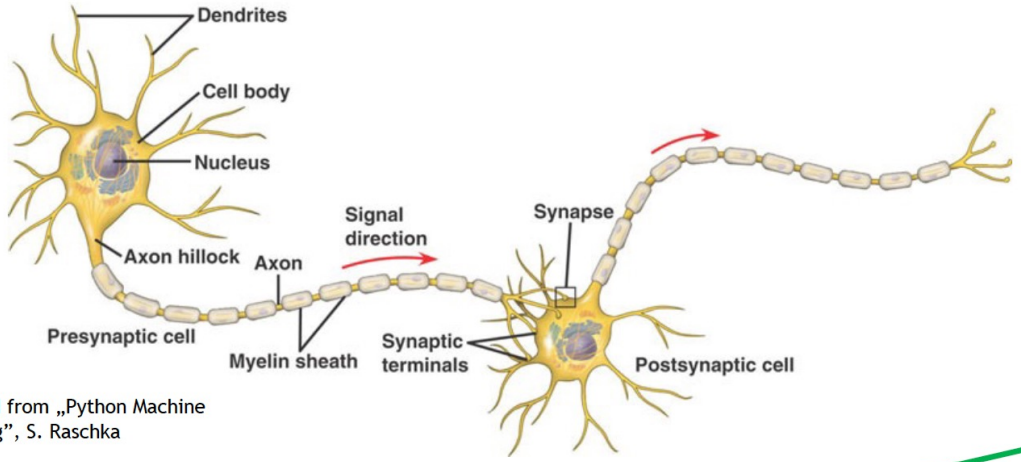
Reinforcement Learning

- 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

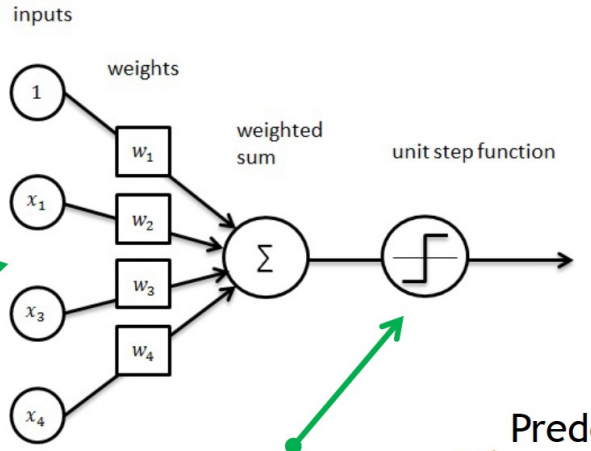
Learning algorithm and Artificial neuron or Perceptron

- 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 $w_j = w_j + \Delta w_j$

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

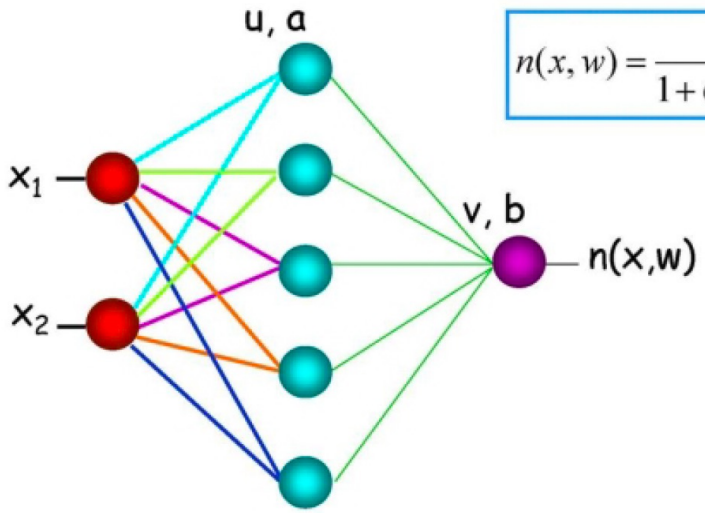
- The second formula is called **perceptron learning rule**, and the η is called the learning rate (just a number between 0 and 1)

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.

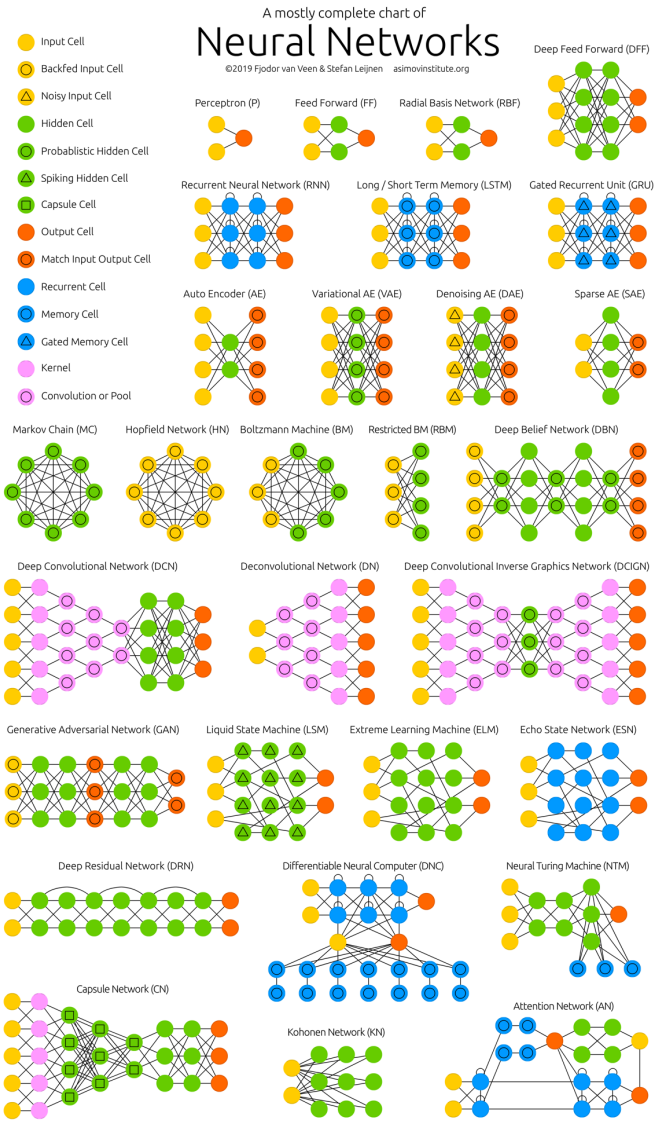
$$f(x,w) = b + \sum_{j=1}^{H} v_j \tanh\left[a_j + \sum_{i=1}^{P} u_{ij} x_i\right]$$

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



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.

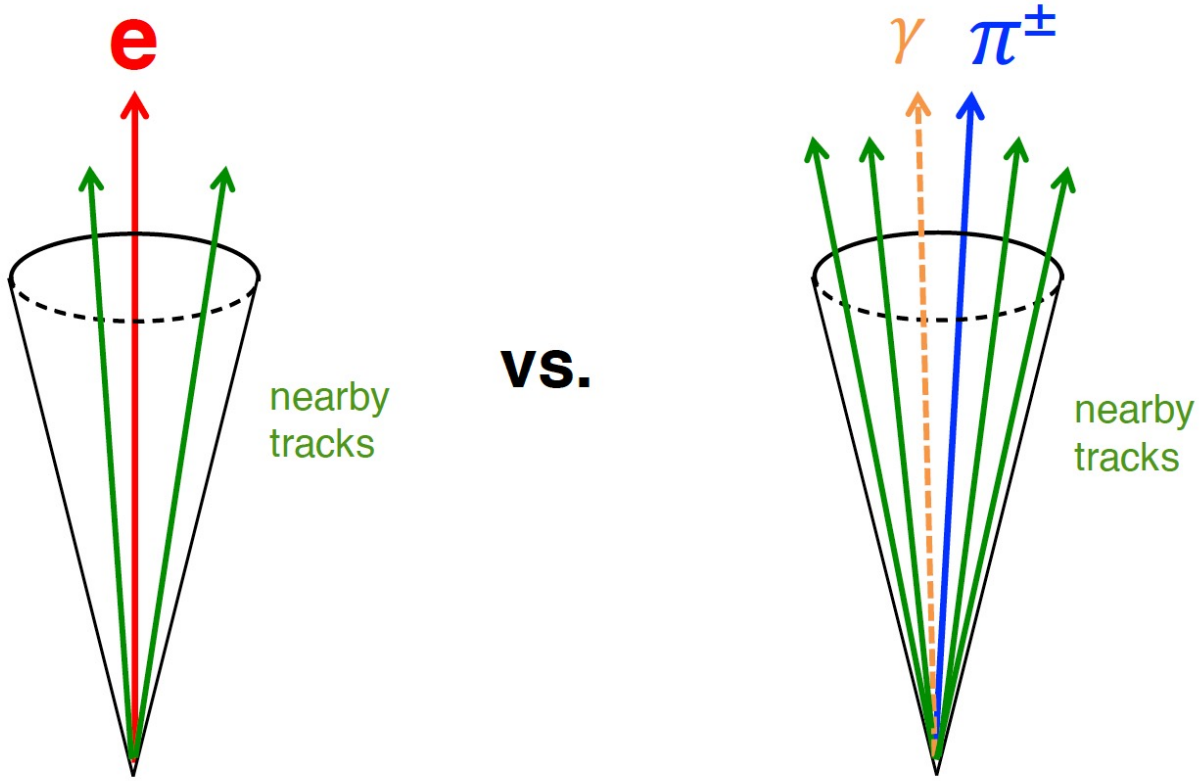
https://www.asimovinstitute.org/neural-network-zoo/

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

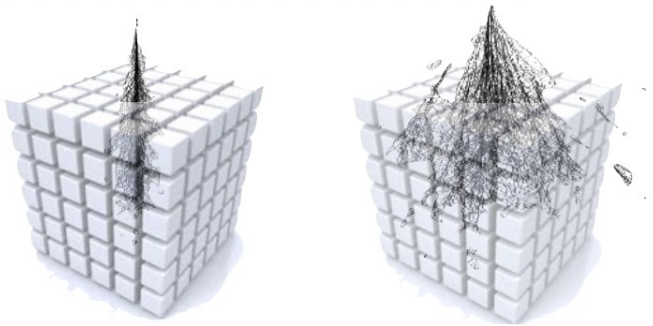
1. **Cut-based selection**
   - Apply requirements on human-designed **features**

machine learning

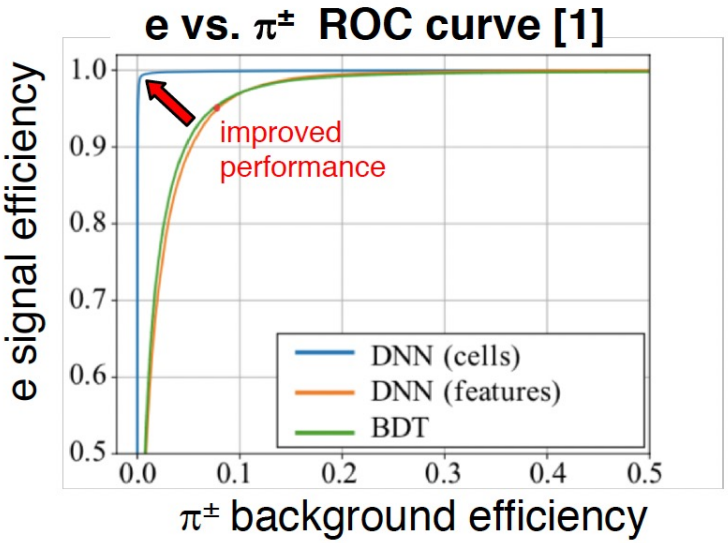
2. **Multi-Variate Algorithms (MVA)**
   - Combine features using *neural networks*, *boosted decision trees*, *likelihoods, etc.*
   - **Exploit *correlations* between features**

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



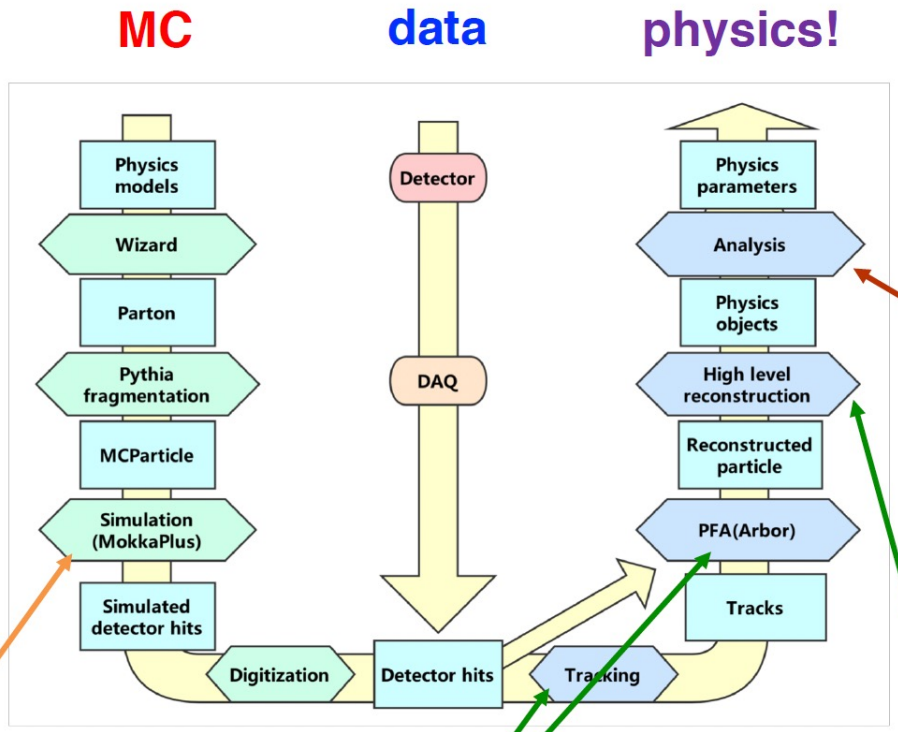
electromagnetic shower hadronic shower



[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)

Use Cases at Colliders

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

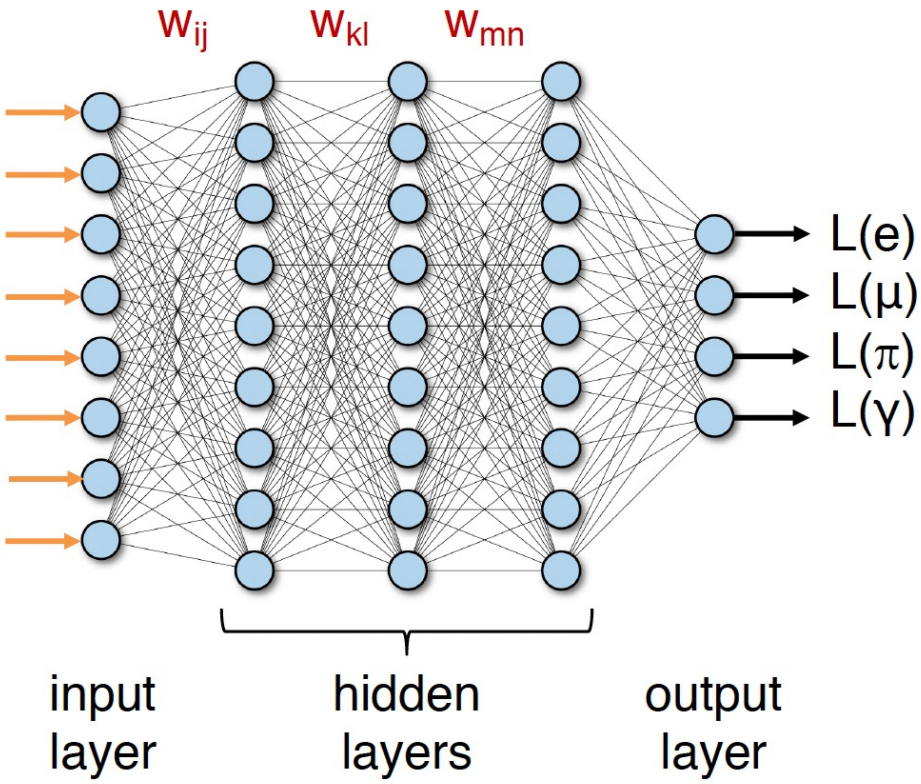
- **Fully-Connected Networks (FCN)**
  - Multiple layers of **fully inter-connected neurons** with variable weights
  - Structure-agnostic → widely applicable

inputs can be…

features

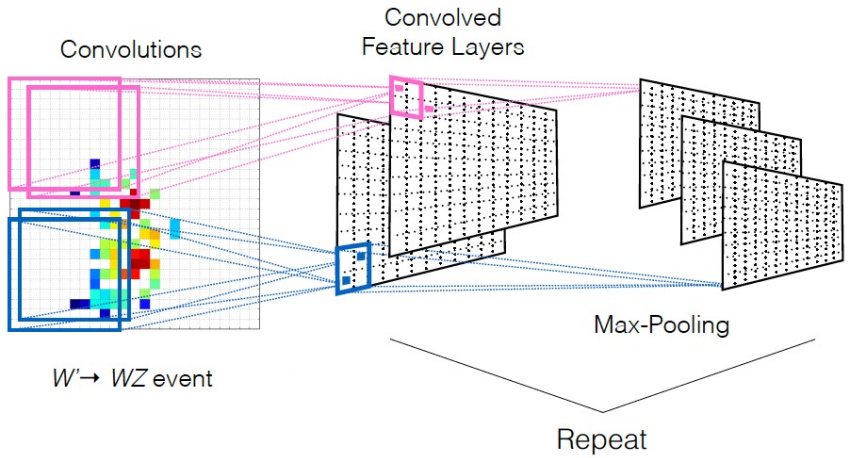
or

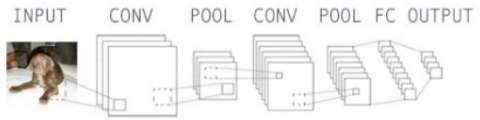
low-level data
(calo cells, track / cluster / particle flow p4's, etc.)



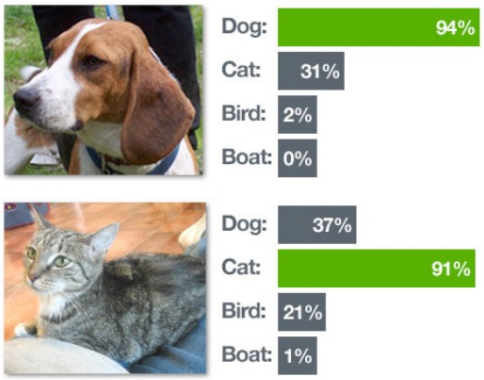
Neural Network Architectures

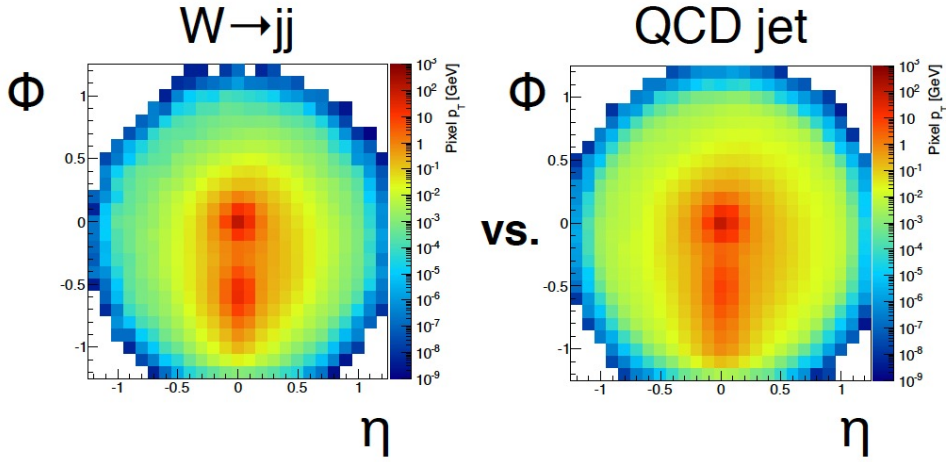
- **Fully-Connected Networks (FCN)**
  - Multiple layers of **fully inter-connected neurons** with variable weights
  - Structure-agnostic → widely applicable
- **Convolutional Neural Networks (CNN)**
  - Specialized layers ("convolutional filters") identify structures at different scales
  - **Computer vision / imaging** applications
  - Assumes **fixed-length** input data





exploits extensive computer vision R&D

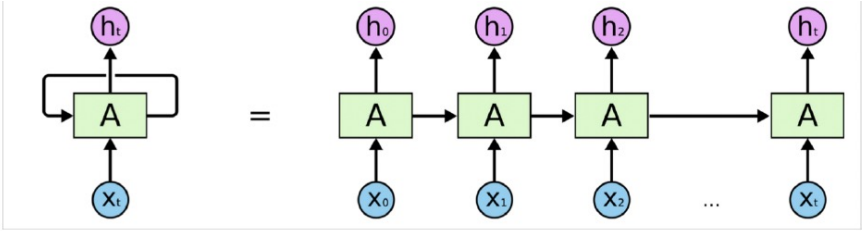




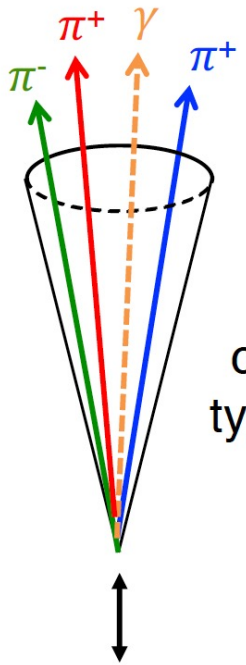
[1] de Oliveira, Kagan, Mackey, Nachmann, Schwartzman, "Jet Images – Deep Learning Edition", JHEP07 (2016) 069

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

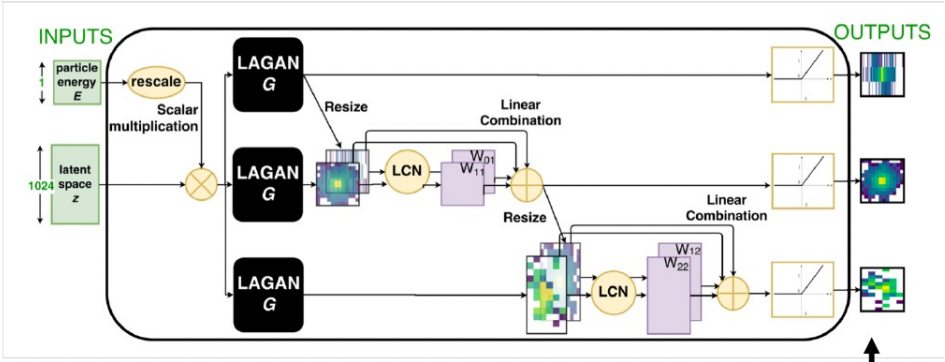
"pm_pt3.5_eta1.1_phi0.2 pp_pt5.6_eta0.3_phi1.8 g_pt10.5_eta1.4_phi0.3 pp_pt3.5_eta1.1_phi1.2."

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**

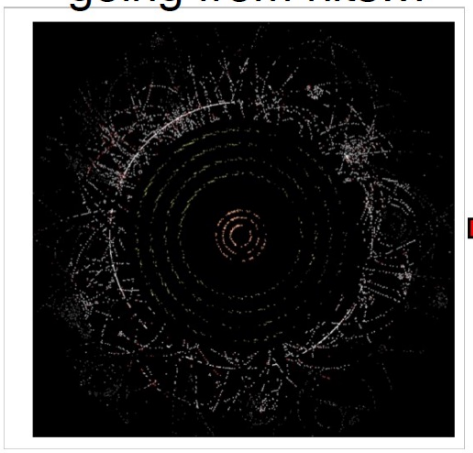
- objects: jet and lepton energies and angles
- events: total / hadronic / missing energy, m_H

fast simulation

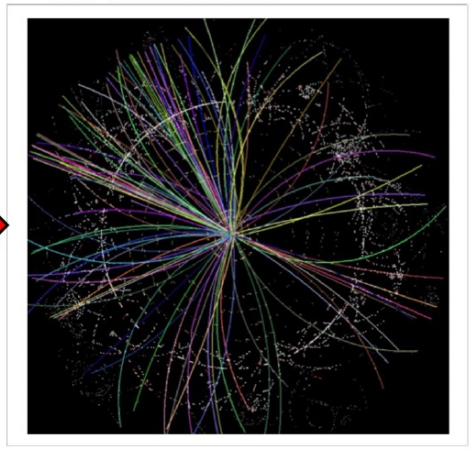
e.g. particle showers in calorimeters

Tracking with ML

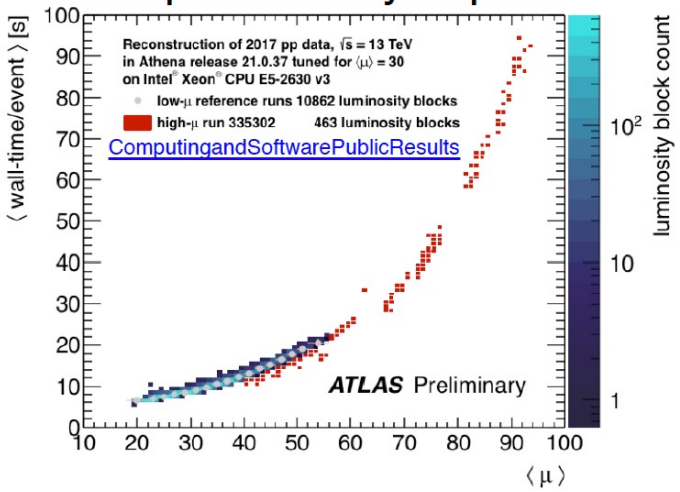
going from hits…



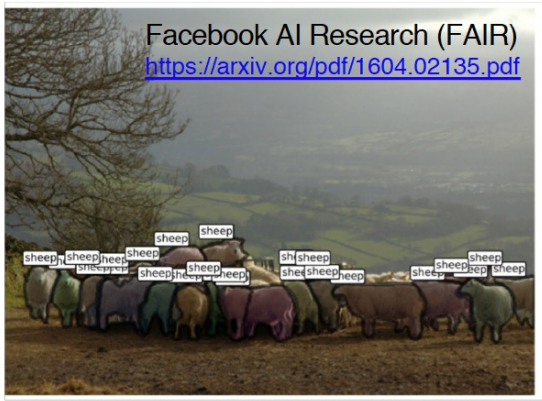
to tracks…

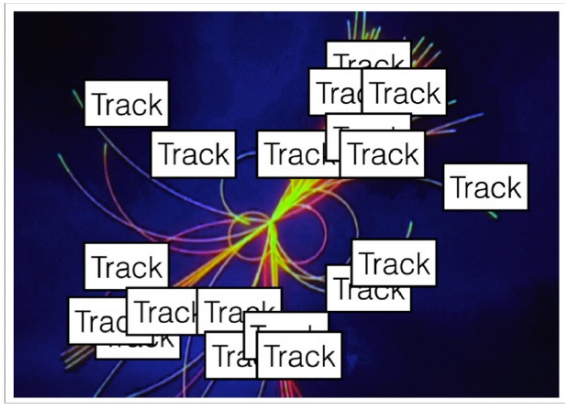


is computationally expensive:



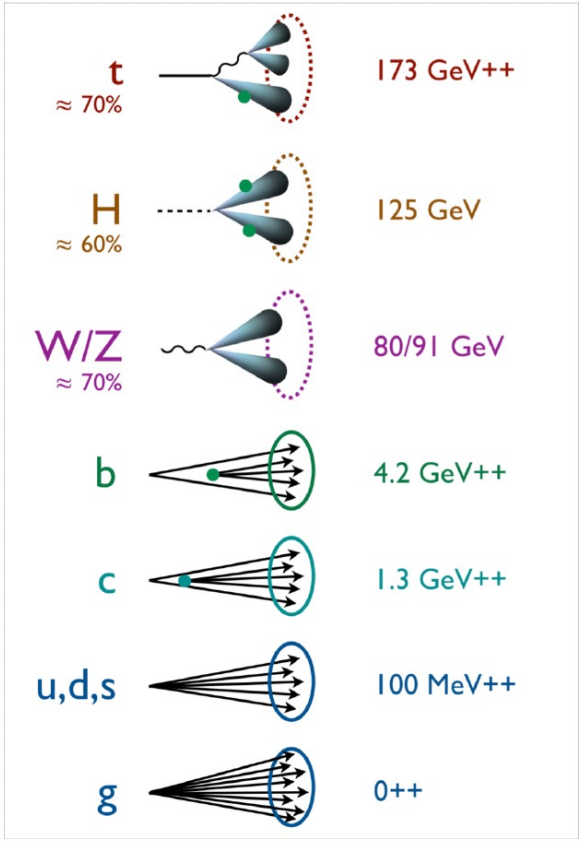
- Major challenge for HL-LHC and future hadron colliders!
- Can leverage **unsupervised learning** techniques to group hits into tracks
- Subject of TrackML challenge



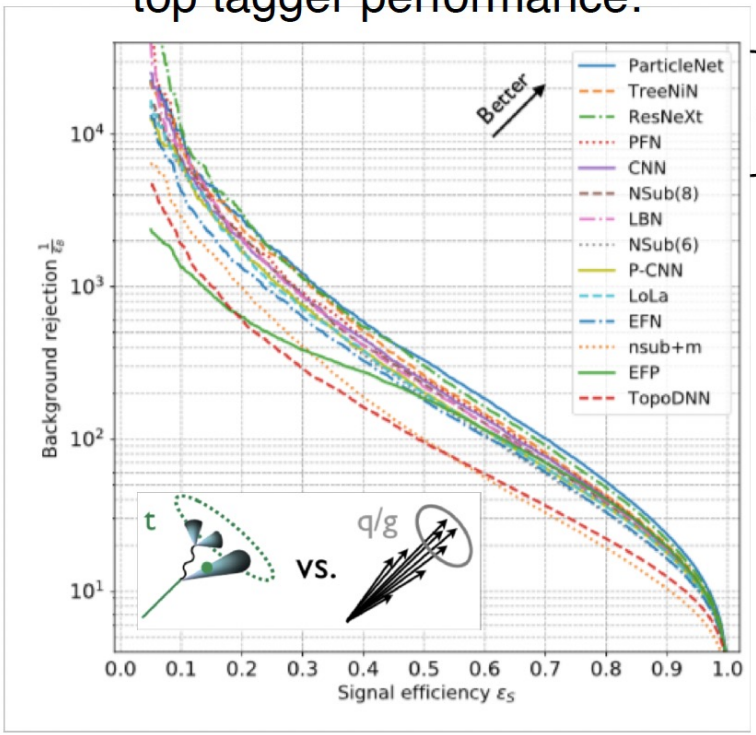


Jet classification with ML

++ = mass from QCD radiation



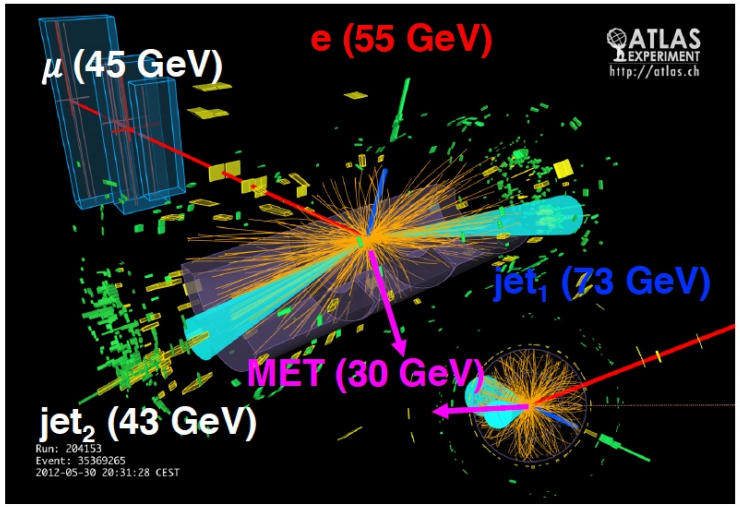
top tagger performance:

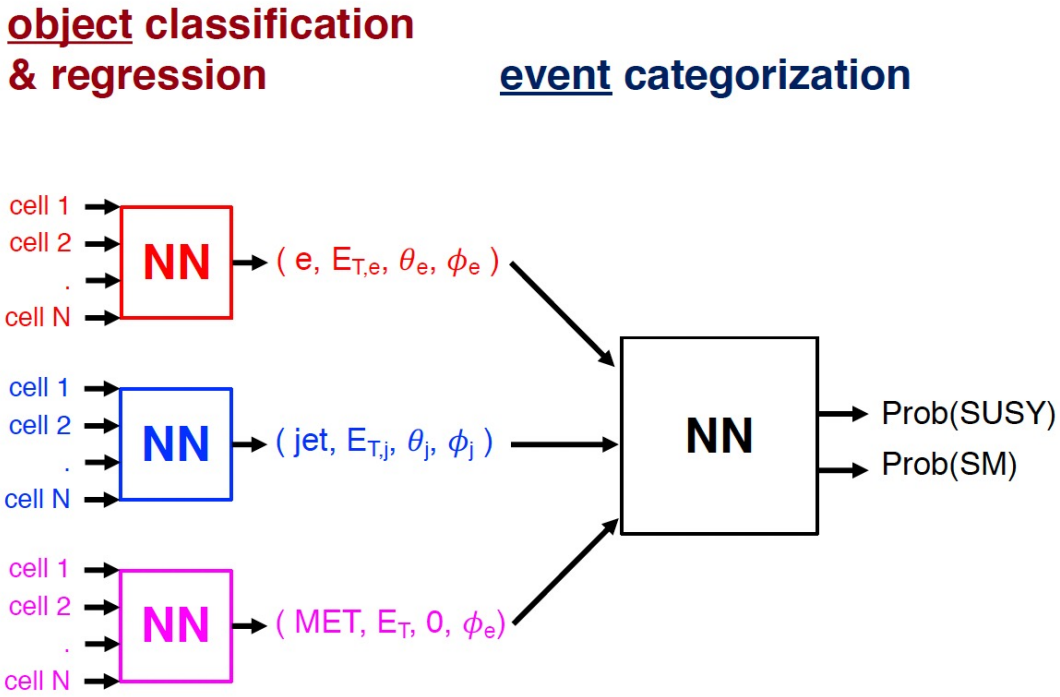


- Deep learning approach often provides best performance for jet classification tasks

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

Strategy for ML event classification

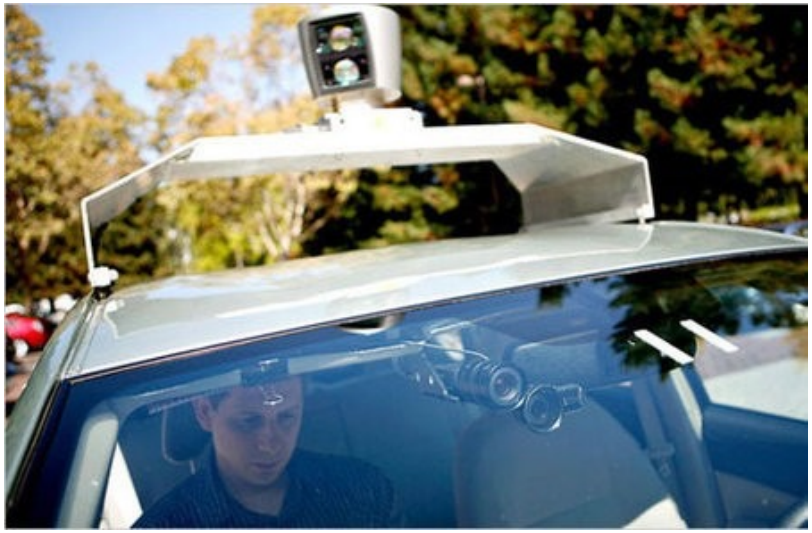


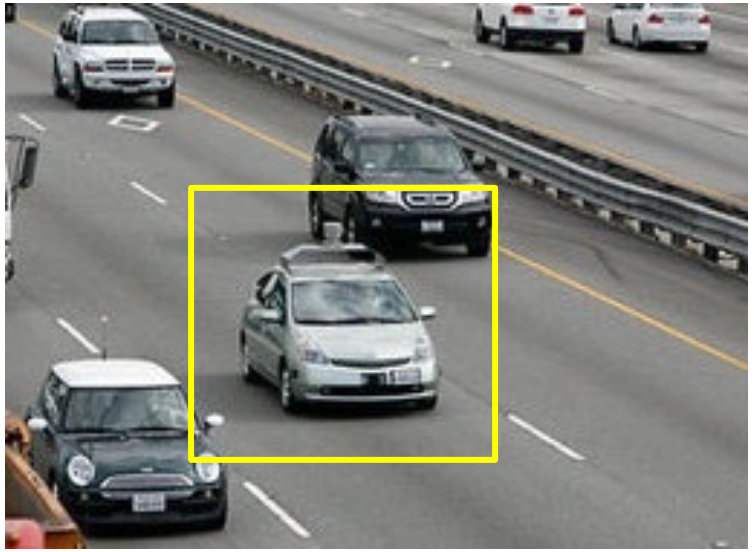


- Factorize the problem: **object tagging** + **event classification**
  - Use **cells** to *classify type* and *measure p4's* of physics **objects** (e, μ, τ, γ, j, MET)
  - Use **object types and p4's** to *categorize* **events** (e.g. SM vs. SUSY) with e.g. RNNs

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

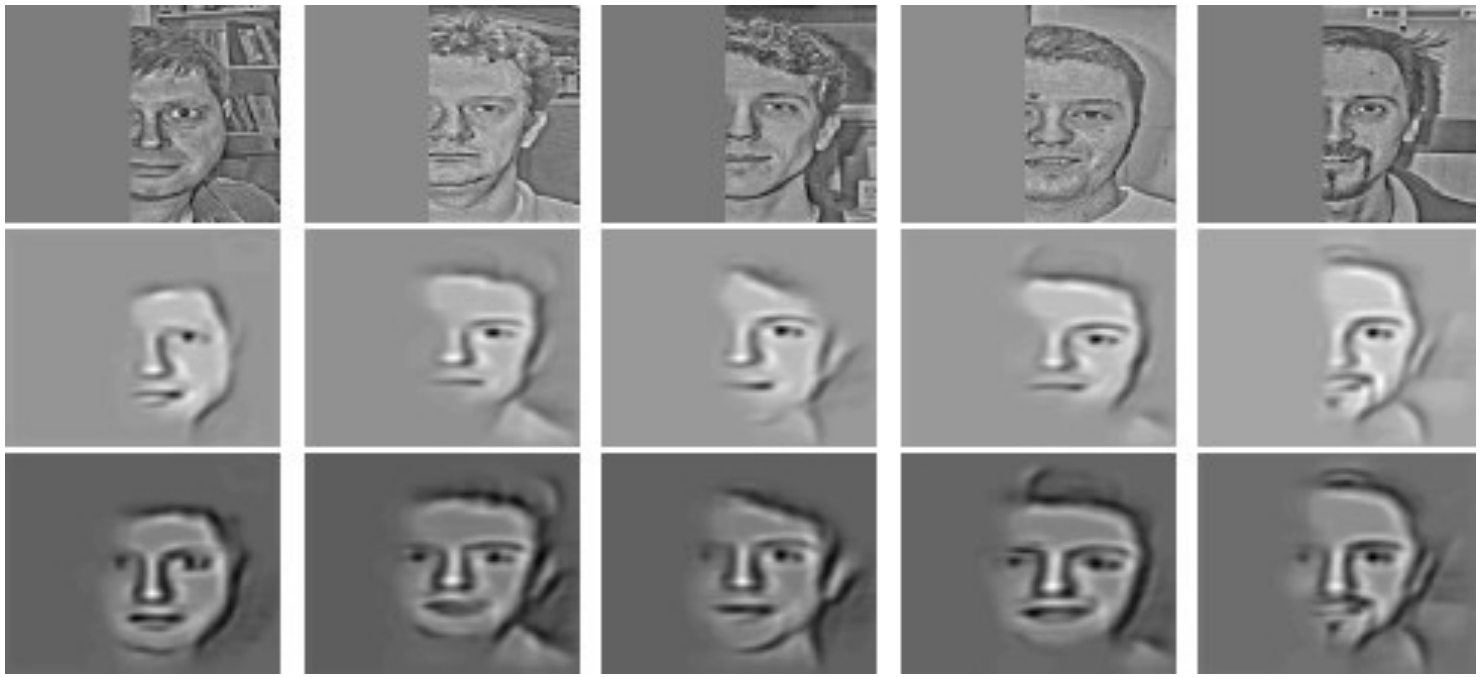
Inference from Deep Learned models

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

Input images

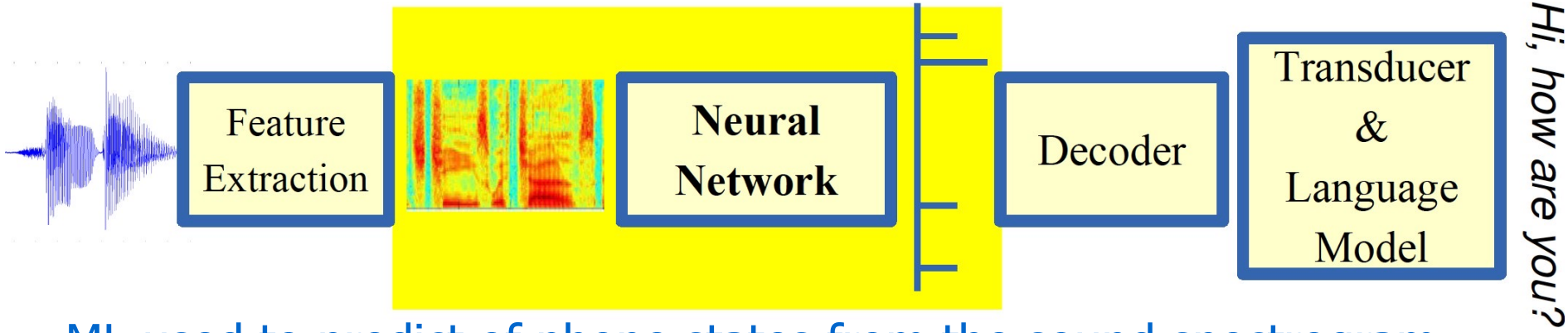
Samples from feedforward Inference (control)

Samples from Full posterior inference

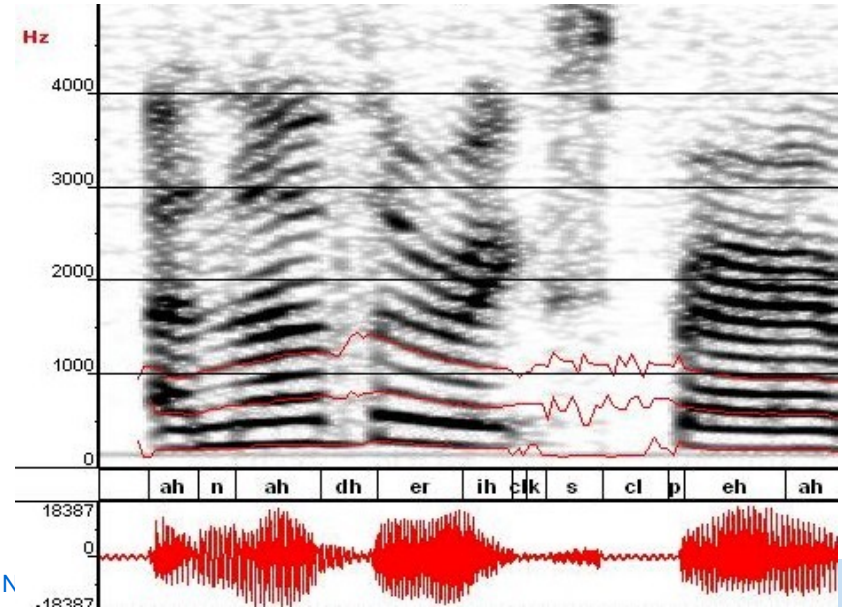


Machine Learning in Automatic Speech Recognition

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]

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 LHC 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