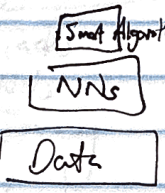
Fudan Summer School Lectures on ML4HEP

August 2026

Motivations: AI/ML having a huge breakthrough moment right now!

LLMs, built from NNs, and trained on enormous amounts of (text) data, are exponentially expanding in capabilities → approaching AGI?

Modern ML:



This has given us AI (LLMs, agents) but it can also do much, much more!

HEP, fundamental physics → enormous data (LHC, neutrinos, astro/cosmo, ...) advantage → also simulations → well-understood (SM, BSM, ...)

↓

ML4HEP: an unprecedented opportunity!

like a telescope/microscope for data!

see deeper & farther into data

better & new measurements, more discovery potential

faster ~~real~~ simulations

for theorists: new methods, proof of concept work

lots of room for creativity & innovation!

- Not "just" for data analysis — also for theory too increasingly
  eg: ML for simplification, Feynman loop integral reduction
  my recent work — also a theoretical "data"!
  scrambled/unscrambled expressions
- ML: an amazing new toolkit (not just tool)
  ~~benefit from~~ interdisciplinary (benefit from CS research)
  on par with calculus or statistics...

~~ML basics~~: Rough Outline

I. ML basics — Tasks
— NNs
— Training

II. Data Representations — Top Tagging @ LHC example
— Images — CNNs
— Sequences — RNNs
— point clouds — GNNs & Transformers

III. Generative Models

IV. Case studies / Applications to HEP — Fast sim / surrogate models
— Anomaly Detection
— DM density from Gaia

What is ML?

but supes turbocharged

"Learning from data" —⊙ "Glorified curve fitting"

We have some data $x_i \in \mathbb{R}^d$ $i = 1, \dots N$

"features" e.g. p_{T_1}, y_1, ϕ_1 p_{T_2}, y_2, ϕ_2 ⋮ or pixels

↖ # of instances e.g. events @ LHC or images of cats & dogs

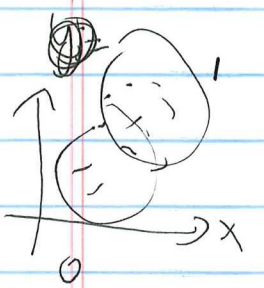
• generally iid from some dist'n

Want to fit a fn $f(x; \theta)$ to the data θ = parameters (or weights) of fn.

Minimize some "loss" or "objective" $L(\theta) = \sum_{i=1}^{N} \ell(f(x_i; \theta))$ wrt parameters θ.

- for example maybe want $f(x_i; \theta) = y_i$ ← target labels e.g. 0 for tops, cats, ... 1 for QCD, dogs, ...

$\ell_f(\theta) = (f(x_i; \theta) - y_i)^2$ "mean squared error"



y_i = discrete "classification"
= continuous "regression"

ML w/ target labels: "supervised ML"

(or any exp't? in HEP, truth labels generally only possible w/ simulation)

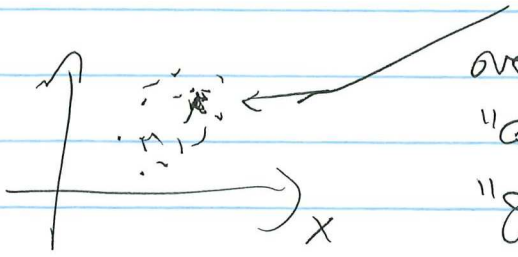
Other kinds of ML ⊙ more data driven

- no labels — unsupervised
- noisy labels — weakly supervised

} less than supervised

Ex:

← "Anomaly detection"
outliers, out of sample



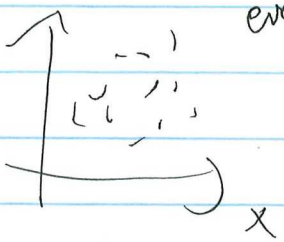
overdensity
"Collective"
"group"

Apps: NP searches
triggering
detector
monitoring

---

events iid,
learn $P(x)$ → "density estimation"

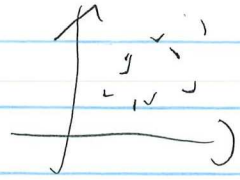
Ex:



one powerful method: "normalizing flows"

Apps: NP searches
phase space — measure $p(x)$ of Galaxies
parameter inference
integration

Ex



learn $P(x)$ (possibly implicitly)
sample from it — "generative models"

- "GANs", "VAEs", flows

Apps: NP searches
surrogate models (eg. GEANT4 calo showers)

Loss terms
for less than supervised
not as obvious — more later!

General principle for loss terms / objective fns:

Maximum Likelihood Estimation (MLE)

maximize: $P(\text{data}|\text{model})$

θ

$\{x_1, x_2, \dots x_N\}$

usually data i.i.d, then

$$P(\text{data}|\text{model}) = \prod_{i=1}^{N} P(x_i|\theta)$$

working w/

$$L = -\log P(\text{data}|\text{model}) = -\sum_{i=1}^{N} \log P(x_i|\theta)$$

MLE has nice properties in limit of large data (unbiased, min variance)

- Ex: regression $f(x,\theta) \to y$

  Suppose y is gaussian distributed around true $\hat{y}(x)$

  $$P(y|x) = Ne^{-\frac{(y-\hat{y}(x))^2}{2\sigma^2}}$$

  our model

  $$P(y|x,\theta) = Ne^{-\frac{(y-f(x,\theta))^2}{2\sigma^2}}$$

Given data $\{x_i, y_i\}$

$$L = -\sum_i \log P(y_i|x_i,\theta) \propto +\sum_i (y_i - f(x_i,\theta))^2$$

MSE loss!

- Another Ex: binary classification

  $f(x;\theta)$ = prob that x is class 1

data: $\{x_1, \dots x_N\}$ class 1 $\{x_{N_1+1} \dots x_{N_1+N_0}\}$ class 0

$$P(\text{data}|\text{model}) = \prod_{i\in 1} f(x_i;\theta) \prod_{j\in 0} (1-f(x_j;\theta))$$

$$L = -\log P = -\sum_{i\in 1} \log f(x_i;\theta) - \sum_{j\in 0} \log(1-f(x_j;\theta))$$

$$= \boxed{-\sum_{i\in \text{data}} y_i \log f(x_i;\theta) + (1-y_i)\log(1-f(x_i;\theta))}$$

"binary cross entropy" — used for binary classification

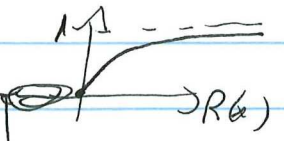
Side note about bce: what minimizes it?

$$L = \int dx \left(P_1(x) \log f(x;\theta) + P_0(x) \log(1-f(x;\theta)) \right)$$

($P_1(x)$: prob density if signal; $P_0(x)$: prob density of bg)

$$\delta L = \frac{P_1(x)}{F(x;\theta)} - \frac{P_0(x)}{1-f(x;\theta)} \Rightarrow \boxed{f(x;\hat{\theta}) = \frac{P_1(x)}{P_1(x)+P_0(x)}}$$

$$R(x) \equiv \frac{P_1(x)}{P_0(x)} \quad \text{"Likelihood ratio"} \qquad = \frac{R(x)}{1+R(x)}$$



So optimal classifier is (monotonic with) LR

"Neyman-Pearson lemma"