II. Come può Machine Learning aiutaci?

Ci sono tanti dati disponibili

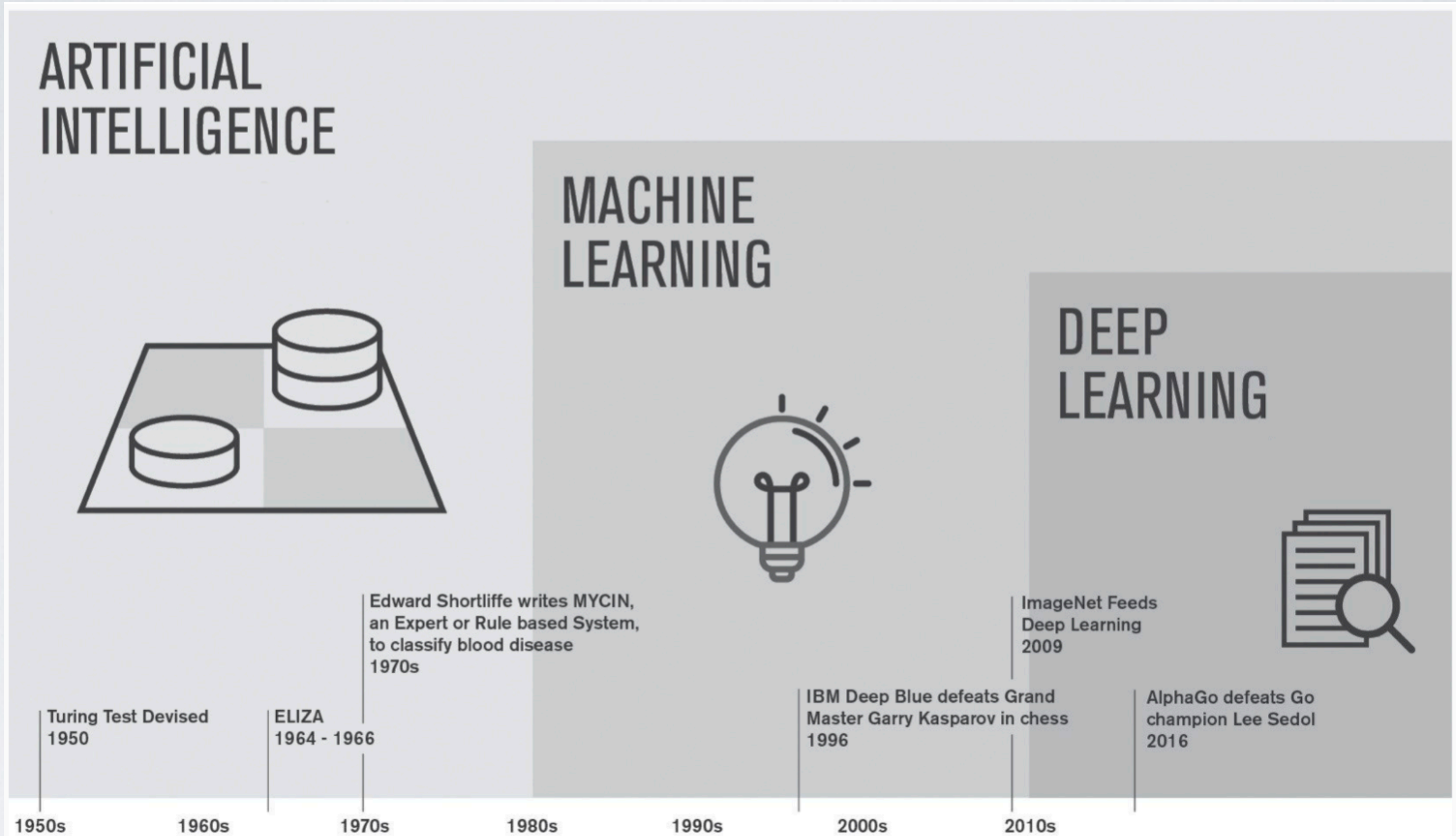
Optimisation delle simulazioni

Trovare i migliori osservabili

Ci sono tanti algoritmi avanzati disponibili

Chi ha usato il ML prima?

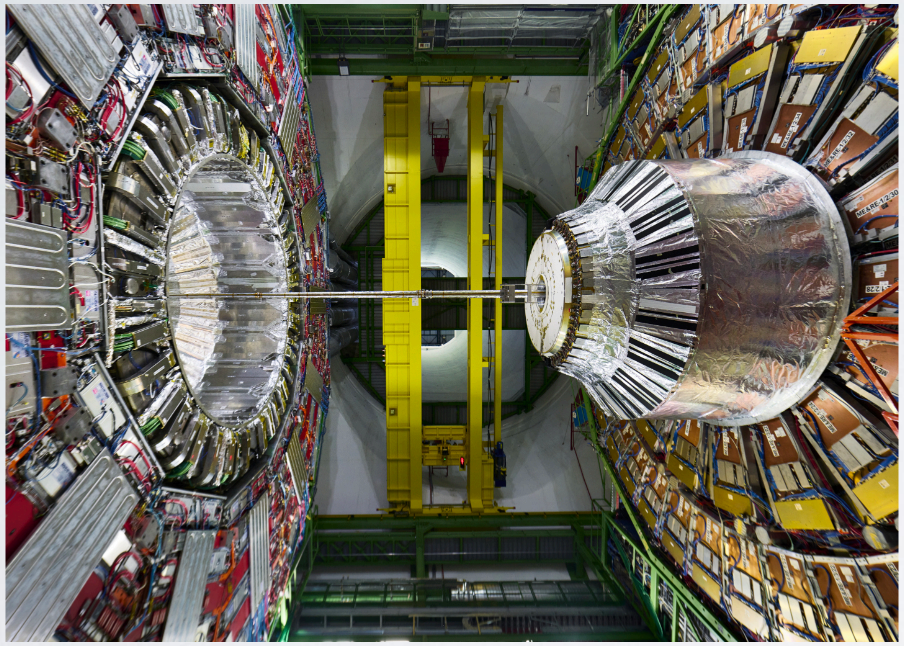
ML algorithms have been around for a while



also in HEP

e.g. Finding Gluon Jets With a Neural Trigger, Lonnblad, Petersen, Rognvaldsson (1990)

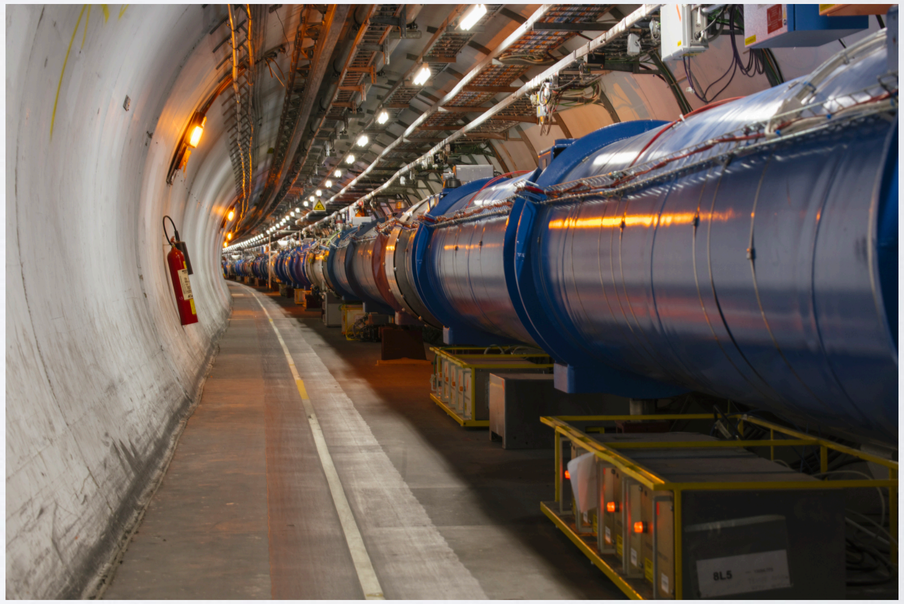
ML algorithms widely used by HEP experiments



protecting hardware to avoid equipment failures

tuning hardware triggers

pile-up



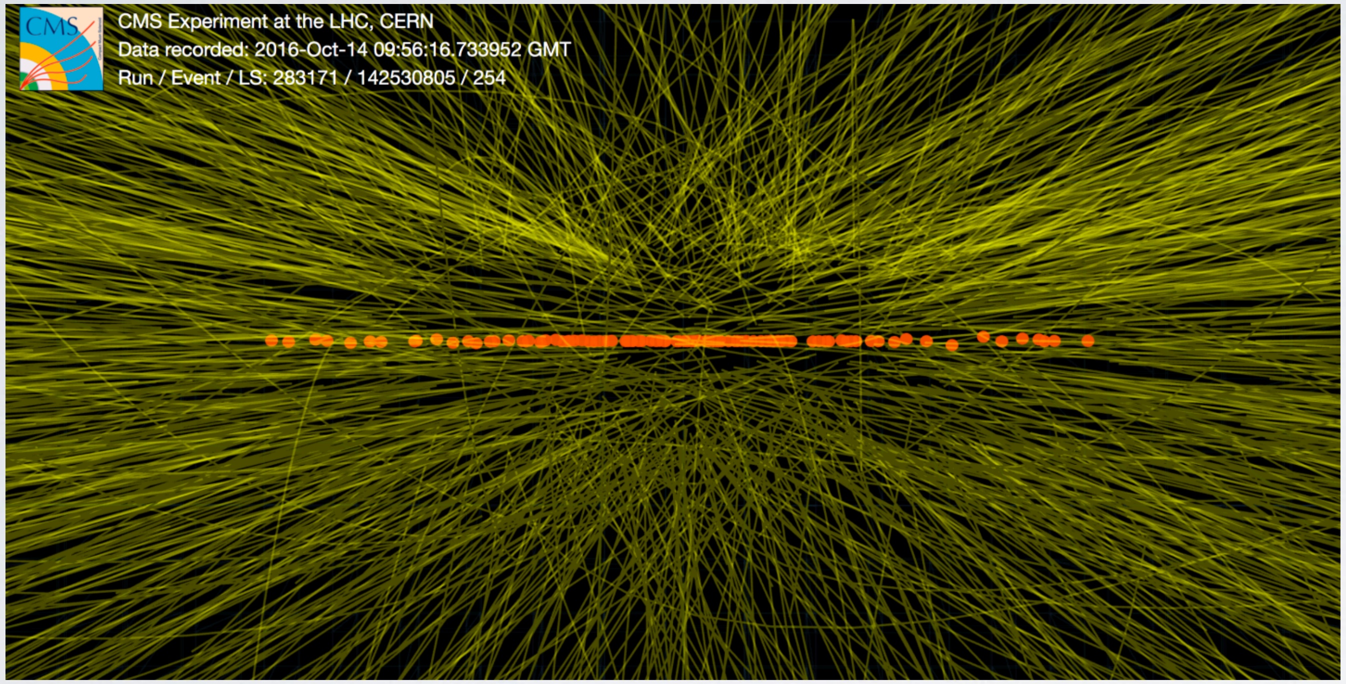
optimising beam quality

particle identification

calorimetry

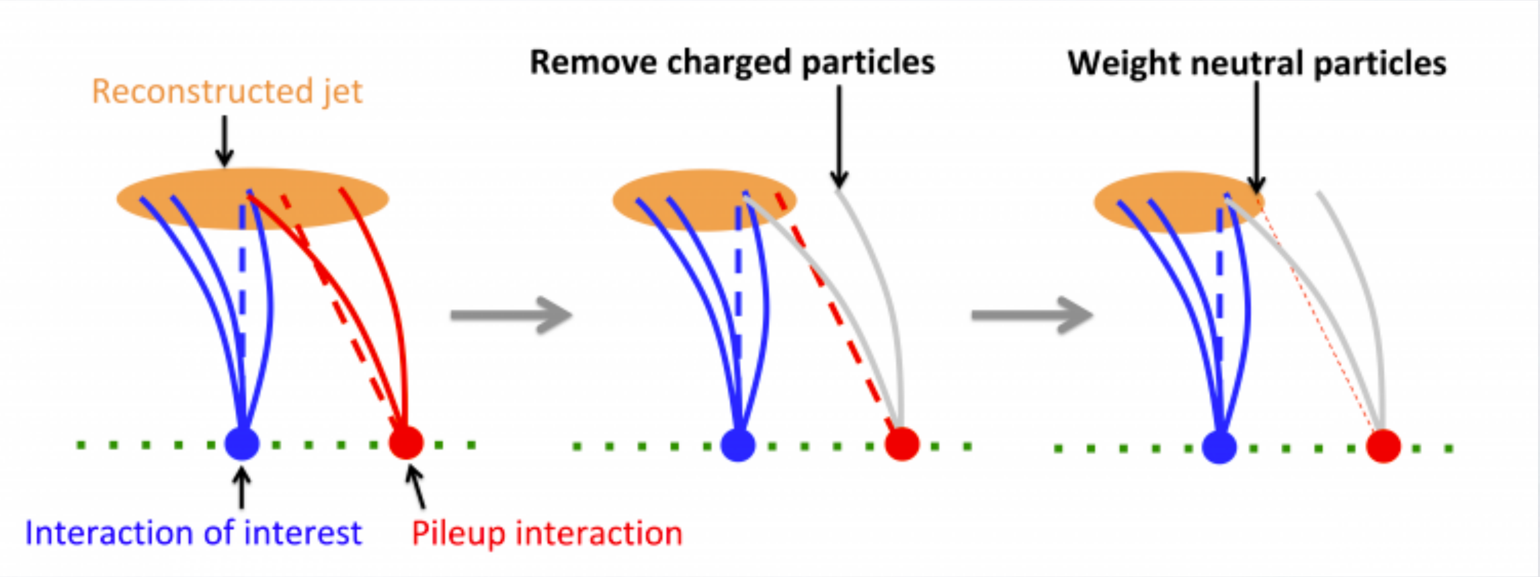
event analysis
(Boosted decision trees - BDT)

Example: Pile Up



LHC run II saw an average of 40 (max 80) interactions per bunch crossing!

Interesting high energy interactions can be contaminated by low energy jet activity



Example: Pile Up

Pileup mitigation at the Large Hadron Collider with Graph Neural Networks

J. Arjona Martínez[1], O. Cerri[2], M. Spiropulu[2], JR. Vlimant[2], and M. Pierini[3]

[1] University of Cambridge, Trinity Ln, Cambridge CB2 1TN, UK
[2] California Institute of Technology, 1200 E. California Blvd, Pasadena, CA 91125
[3] CERN, CH-1211 Geneva, Switzerland
e-mail: maurizio.pierini@cern.ch

Abstract. At the Large Hadron Collider, the high transverse-momentum events studied by experimental collaborations occur in coincidence with parasitic low transverse-momentum collisions, usually referred to as pileup. Pileup mitigation is a key ingredient of the online and offline event reconstruction as pileup affects the reconstruction accuracy of many physics observables. We present a classifier based on Graph Neural Networks, trained to retain particles coming from high-transverse-momentum collisions, while rejecting those coming from pileup collisions. This model is designed as a refinement of the `PUPPI` algorithm [1], employed in many LHC data analyses since 2015. Thanks to an extended basis of input information and the learning capabilities of the considered network architecture, we show an improvement in pileup-rejection performances with respect to state-of-the-art solutions.

Example: Pile Up

Pileup mitigation at the Large Hadron Collider with Graph Neural Networks

J. Arjona M

[1] University
[2] California
[3] CERN, CH
e-mail: mau

ep-ph] 13 Jun 2019

Ab
co
as
th
Ne
th
em
lea
pe

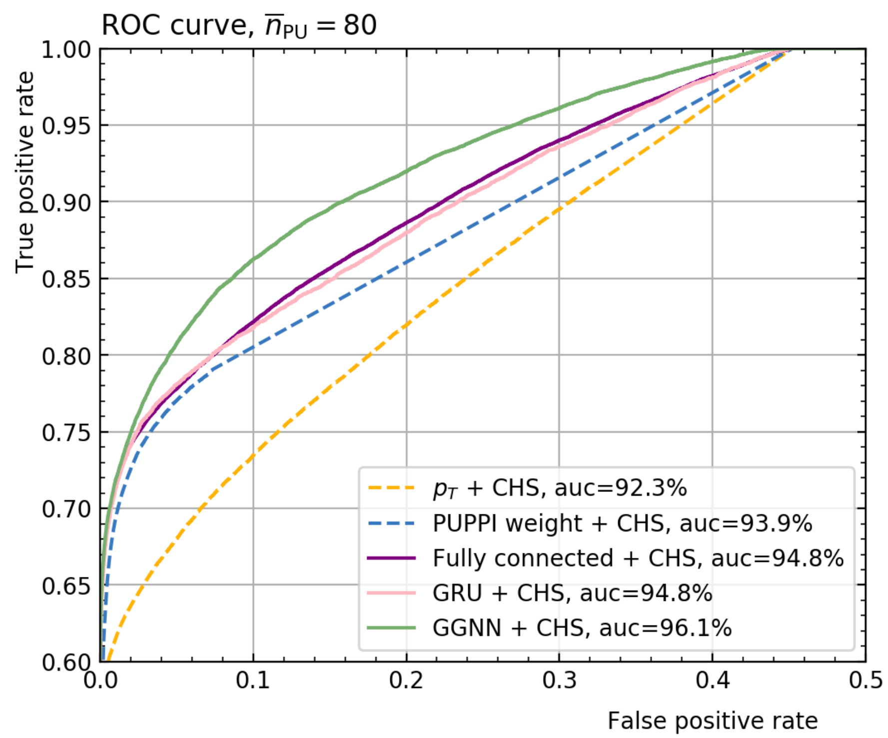


Fig. 3. Receiver Operating Characteristic (ROC) curve for our proposed features and models. Classifiers based on `PUPPI` weight and p_T are included as an indicator of the expected performance of `PUPPI` and *SoftKiller* respectively. GGNN outperform other proposed architectures.

$\overline{n}_{\mathrm{PU}}$	20 (CHS)	80 (CHS)	140 (CHS)	80 (No CHS)
p_T (SoftKiller)	92.3%	92.3%	92.5%	64.9%
`PUPPI` weight	94.1%	93.9%	94.4%	65.1%
Fully-connected	95.0%	94.8%	94.8%	68.5%
GRU	94.8%	94.8%	94.7%	68.8%
GGNN	**96.1%**	**96.1%**	**96.0%**	**70.1%**

Table 1. Area under the curve for the different discriminating variables and models. The highest values, highlighted in bold, are obtained when the GGNN architecture is used.

Example: Cut selection with BDTs

e.g. [https://arxiv.org/pdf/physics/0408124]

- Decision trees - multivariate analysis tool.
- Supervised learning classification technique
- Boosting - multilayer binary classifiers
- Alternative to Artificial Neural Networks (ANN)

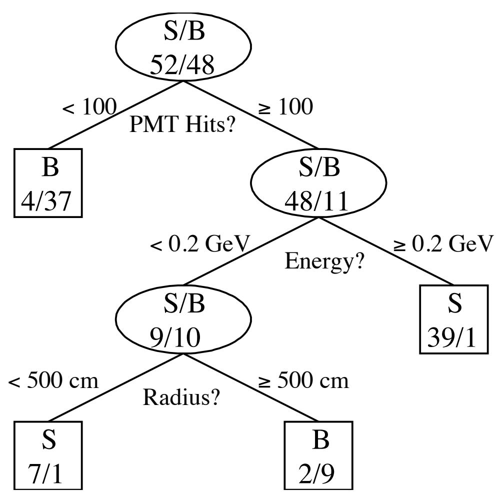
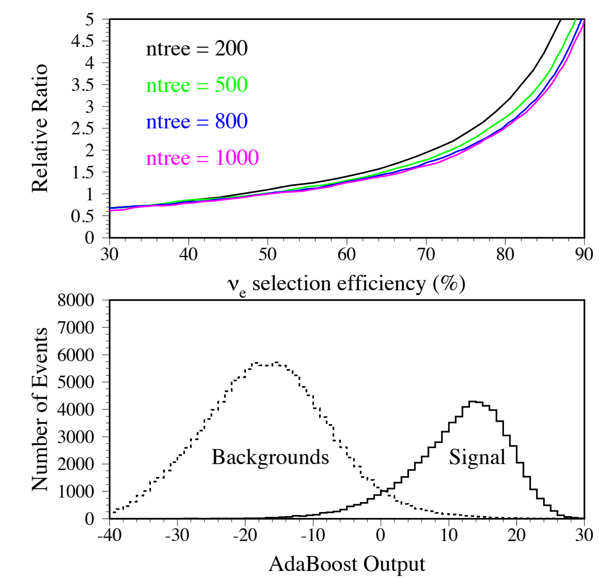
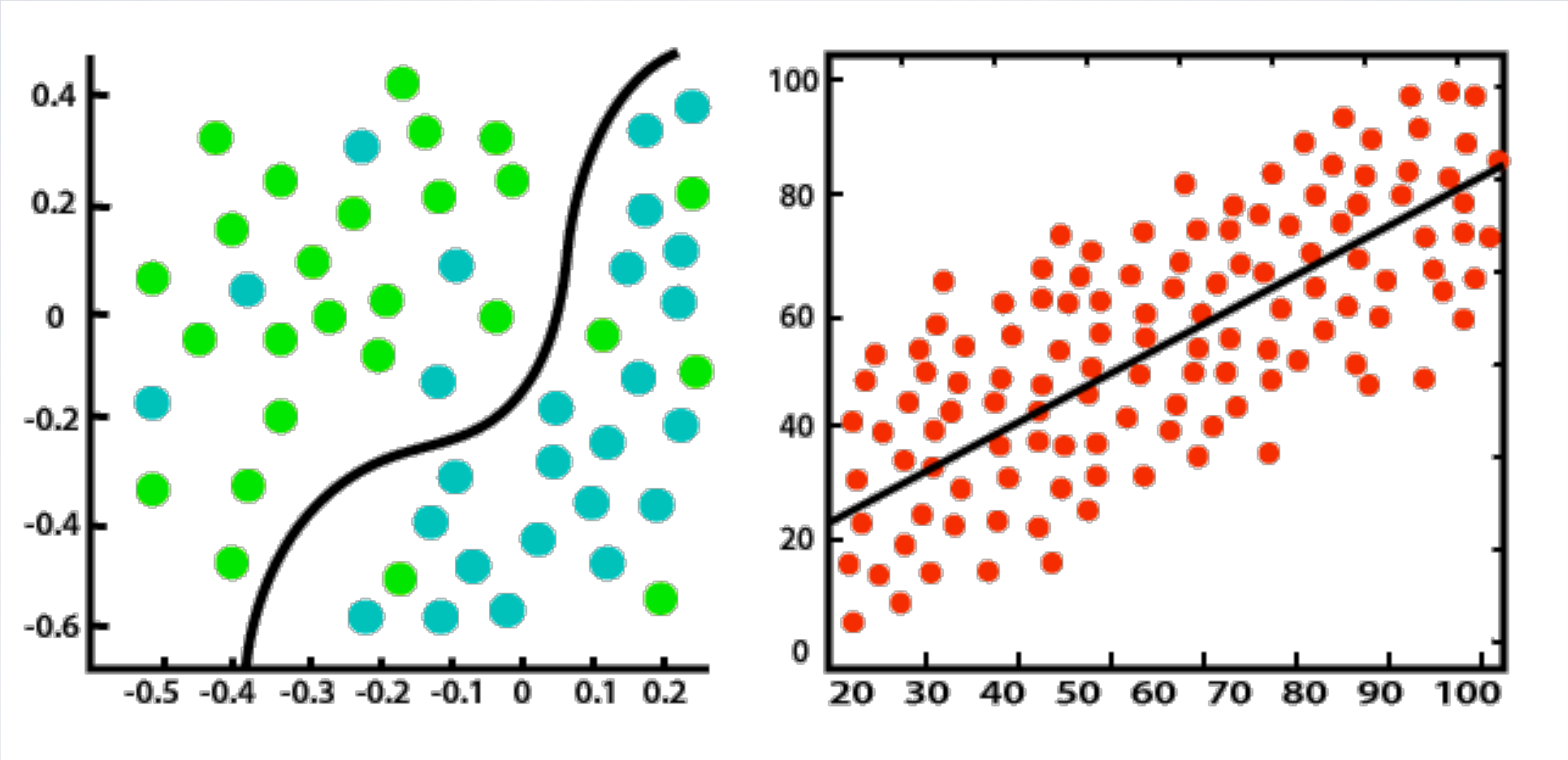


FIG. 1: Schematic of a decision tree. S for signal, B for background. Terminal nodes(called leaves) are shown in boxes. If signal events are dominant in one leave, then this leave is signal leave; otherwise, background leave.

Q: Signal or Background?



Supervised Learning

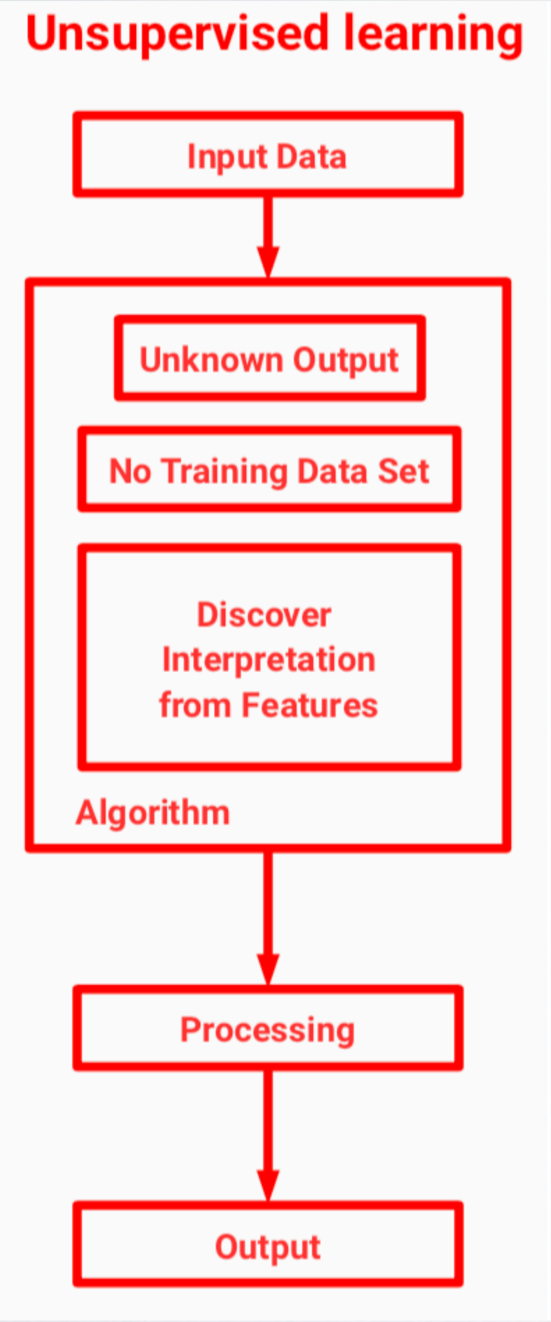


Classification

Regression

"Learn from examples"

Unsupervised Learning



unlabelled data

find patterns

"let the data guide you…"

the world of ML techniques can be a bit intimidating

ML in HEP

many examples collected in the Living Review: https://github.com/iml-wg/HEPML-LivingReview

README.md

A Living Review of Machine Learning for Particle Physics

Modern machine learning techniques, including deep learning, is rapidly being applied, adapted, and developed for high energy physics. The goal of this document is to provide a nearly comprehensive list of citations for those developing and applying these approaches to experimental, phenomenological, or theoretical analyses. As a living document, it will be updated as often as possible to incorporate the latest developments. A list of proper (unchanging) reviews can be found within. Papers are grouped into a small set of topics to be as useful as possible. Suggestions are most welcome.

download review

The purpose of this note is to collect references for modern machine learning as applied to particle physics. A minimal number of categories is chosen in order to be as useful as possible. Note that papers may be referenced in more than one category. The fact that a paper is listed in this document does not endorse or validate its content - that is for the community (and for peer-review) to decide. Furthermore, the classification here is a best attempt and may have flaws - please let us know if (a) we have missed a paper you think should be included, (b) a paper has been misclassified, or (c) a citation for a paper is not correct or if the journal information is now available. In order to be as useful as possible, this document will continue to evolve so please check back before you write your next paper. If you find this review helpful, please consider citing it using \cite{hepmllivingreview} in HEPML.bib.

- strange jets
  - Strange Jet Tagging
  - A tagger for strange jets based on tracking information using long short-term memory [DOI]
  - Maximum performance of strange-jet tagging at hadron colliders
- b-tagging
  - Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV [DOI]
  - Jet Flavor Classification in High-Energy Physics with Deep Neural Networks [DOI]
  - Identifying Heavy-Flavor Jets Using Vectors of Locally Aggregated Descriptors [DOI]
  - Jet Flavour Classification Using DeepJet [DOI]
  - Identification of Jets Containing b-Hadrons with Recurrent Neural Networks at the ATLAS Experiment
  - Deep Sets based Neural Networks for Impact Parameter Flavour Tagging in ATLAS
- Flavor physics
  - 'Deep' Dive into $b \to c$ Anomalies: Standardized and Future-proof Model Selection Using Self-normalizing Neural Networks
- BSM particles and models
  - Automating the Construction of Jet Observables with Machine Learning [DOI]
  - Searching for Exotic Particles in High-Energy Physics with Deep Learning [DOI]
  - Interpretable deep learning for two-prong jet classification with jet spectra [DOI]
  - A deep neural network to search for new long-lived particles decaying to jets [DOI]
  - Fast convolutional neural networks for identifying long-lived particles in a high-granularity calorimeter [DOI]
  - Casting a graph net to catch dark showers [DOI]

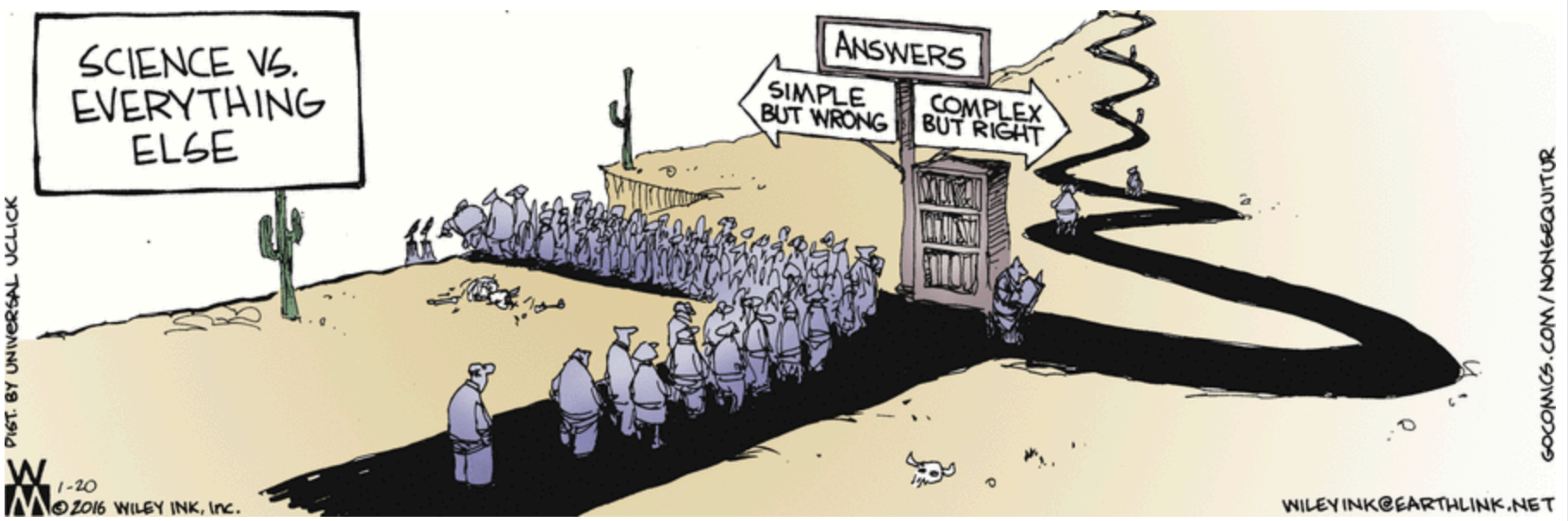
- Classification
  - Parameterized classifiers
    - Parameterized neural networks for high-energy physics [DOI]
    - Approximating Likelihood Ratios with Calibrated Discriminative Classifiers
    - E Pluribus Unum Ex Machina: Learning from Many Collider Events at Once
    - Jet images
      - How to tell quark jets from gluon jets
      - Jet-Images: Computer Vision Inspired Techniques for Jet Tagging [DOI]
      - Playing Tag with ANN: Boosted Top Identification with Pattern Recognition [DOI]
      - Jet-images — deep learning edition [DOI]
      - Quark versus Gluon Jet Tagging Using Jet Images with the ATLAS Detector
      - Boosting $H\to b\bar b$ with Machine Learning [DOI]
      - Learning to classify from impure samples with high-dimensional data [DOI]
      - Parton Shower Uncertainties in Jet Substructure Analyses with Deep Neural Networks [DOI]
      - Deep learning in color: towards automated quark/gluon [DOI]
      - Deep-learning Top Taggers or The End of QCD? [DOI]
      - Pulling Out All the Tops with Computer Vision and Deep Learning [DOI]
      - Reconstructing boosted Higgs jets from event image segmentation
      - An Attention Based Neural Network for Jet Tagging
      - Quark-Gluon Jet Discrimination Using Convolutional Neural Networks [DOI]
      - Learning to Isolate Muons
      - Deep learning jet modifications in heavy-ion collisions
      - Identifying the Quantum Properties of Hadronic Resonances using Machine Learning
    - Event images
      - Topology classification with deep learning to improve real-time event selection at the LHC [DOI]
      - Convolutional Neural Networks with Event Images for Pileup Mitigation with the ATLAS Detector
      - Boosting $H\to b\bar b$ with Machine Learning [DOI]
      - End-to-End Physics Event Classification with the CMS Open Data: Applying Image-based Deep Learning on Detector Data to Directly Classify Collision Events at the LHC [DOI]
      - Disentangling Boosted Higgs Boson Production Modes with Machine Learning

Processing data with Pandas

exercise: `pandas_textprocessing.ipynb`

- read in list of papers
- search of most frequent words
- order by date? trends?

Word	Frequency
learning	290
neural	181
machine	168
physics	164
networks	164
deep	125
jet	112
particle	88
energy	80
network	60


SCIENCE VS.
EVERYTHING
ELSE
ANSWERS
SIMPLE
BUT WRONG
COMPLEX
BUT RIGHT
DIST. BY UNIVERSAL UCLICK
1-20
© 2016 WILEY INK, Inc.
GOCOMICS.COM/NONSEQUITUR
WILEYINK@EARTHLINK.NET

ML for theory

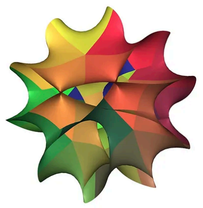
theory problems often involve:

- analytic understanding of an idealised system
- well defined approximations with an exact solution
- symbolic processing

some problems have obvious ML applications:
e.g. searching the string moduli space

Characterising Universes in String Theory using Geometric Learning

Dr Challenger Mishra



Abstract

One of the holy grails of modern theoretical physics is the unification of Quantum Mechanics with Einstein's relativity. String theory is the only known consistent theory of quantum gravity, and arguably the most promising candidate for a unified theory of physics. Since its inception in the late 1960s, it has provided tremendous insights into our understanding of the physical world, and has overseen many interesting developments in various branches of pure mathematics and theoretical physics. Despite string theory's many successes, a string model that explains all observed data from cosmology and particle physics experiments, has eluded discovery. This is owing to the particularly large landscape of valid string theory solutions, estimated to be of the size 10^{270,000}. Most of these solutions are thought to lead to descriptions of universes that do not resemble ours in detail.

$$\int \frac{d^{4-2\epsilon}k_1 d^{4-2\epsilon}k_2}{(k_1^2 - m_1^2)(k_2^2 - m_2^2)((k_1+k_2)^2 - m_3^2)}$$

others are less obvious
how can a simplify a complicated algebraic expression?
solving differential equations analytically (e.g. Feynman integrals)

symbolic processing with ML

some experts are thinking in this direction: https://arxiv.org/pdf/1912.01412.pdf

DEEP LEARNING FOR SYMBOLIC MATHEMATICS

Guillaume Lample*
Facebook AI Research
glample@fb.com

François Charton*
Facebook AI Research
fcharton@fb.com

ABSTRACT

Neural networks have a reputation for being better at solving statistical or approximate problems than at performing calculations or working with symbolic data. In this paper, we show that they can be surprisingly good at more elaborated tasks in mathematics, such as symbolic integration and solving differential equations. We propose a syntax for representing mathematical problems, and methods for generating large datasets that can be used to train sequence-to-sequence models. We achieve results that outperform commercial Computer Algebra Systems such as Matlab or Mathematica.

$$162x\log(x)y' + 2y^3\log(x)^2 - 81y\log(x) + 81y = 0 \qquad y = \frac{9\sqrt{x}\sqrt{\frac{1}{\log(x)}}}{\sqrt{c+2x}}$$

In Table 5, we report the top 10 hypotheses returned by our model for this equation. We observe that all generations are actually valid solutions, although they are expressed very differently. They are however not all equal: merging the square roots within the first and third equations would give the same expression except that the third one would contain a factor 2 in front of the constant c, but up to a change of variable, these two solutions are actually equivalent. The ability of the model to recover equivalent expressions, without having been trained to do so, is very intriguing.

Hypothesis	Score	Hypothesis	Score
$\frac{9\sqrt{x}\sqrt{\frac{1}{\log(x)}}}{\sqrt{c+2x}}$	-0.047	$\frac{9}{\sqrt{\frac{c\log(x)}{x}+2\log(x)}}$	-0.124
$\frac{9\sqrt{x}}{\sqrt{c+2x}\sqrt{\log(x)}}$	-0.056	$\frac{9\sqrt{x}}{\sqrt{c\log(x)+2x\log(x)}}$	-0.139
$\frac{9\sqrt{2}\sqrt{x}\sqrt{\frac{1}{\log(x)}}}{2\sqrt{c+x}}$	-0.115	$\frac{9}{\sqrt{\frac{c}{x}+2}\sqrt{\log(x)}}$	-0.144
$9\sqrt{x}\sqrt{\frac{1}{c\log(x)+2x\log(x)}}$	-0.117	$9\sqrt{\frac{1}{\frac{c\log(x)}{x}+2\log(x)}}$	-0.205
$\frac{9\sqrt{2}\sqrt{x}}{2\sqrt{c+x}\sqrt{\log(x)}}$	-0.124	$9\sqrt{x}\sqrt{\frac{1}{c\log(x)+2x\log(x)+\log(x)}}$	-0.232

Table 5: Top 10 generations of our model for the first order differential equation $162x\log(x)y' + 2y^3\log(x)^2 - 81y\log(x) + 81y = 0$, generated with a beam search. All hypotheses are valid solutions, and are equivalent up to a change of the variable c. Scores are log-probabilities normalized by sequence lengths.

examples taken from
Matt Schwartz at Amplitudes 2022, Prague

https://indico.cern.ch/event/1101193/contributions/4963682/attachments/2491848/4279288/Amplitudes-2022-Schwartz.pdf

examples taken from
Matt Schwartz at Amplitudes 2022, Prague

ANALYSING MATHEMATICAL REASONING ABILITIES OF NEURAL MODELS

David Saxton
DeepMind
saxton@google.com

Edward Grefenstette
DeepMind
egrefen@fb.com

Felix Hill
DeepMind
felixhill@google.com

Pushmeet Kohli
DeepMind
pushmeet@google.com

ABSTRACT

Mathematical reasoning—a core ability within human intelligence—presents some unique challenges as a domain: we do not come to understand and solve mathematical problems primarily on the back of experience and evidence, but on the basis of inferring, learning, and exploiting laws, axioms, and symbol manipulation rules. In this paper, we present a new challenge for the evaluation (and eventually the design) of neural architectures and similar system, developing a task suite of mathematics problems involving sequential questions and answers in a free-form textual input/output format. The structured nature of the mathematics domain, covering arithmetic, algebra, probability and calculus, enables the construction of training and test splits designed to clearly illuminate the capabilities and failure-modes of different architectures, as well as evaluate their ability to compose and relate knowledge and learned processes. Having described the data generation process and its potential future expansions, we conduct a comprehensive analysis of models from two broad classes of the most powerful sequence-to-sequence architectures and find notable differences in their ability to resolve mathematical problems and generalize their knowledge.

Question: `Solve -42*r + 27*c = -1167 and 130*r + 4*c = 372 for r.`
Answer: `4`
Question: `Calculate -841880142.544 + 411127.`
Answer: `-841469015.544`
Question: `Let x(g) = 9*g + 1. Let q(c) = 2*c + 1. Let f(i) = 3*i - 39. Let w(j) = q(x(j)). Calculate f(w(a)).`
Answer: `54*a - 30`
Question: `Let e(l) = l - 6. Is 2 a factor of both e(9) and 2?`
Answer: `False`
Question: `Let u(n) = -n**3 - n**2. Let e(c) = -2*c**3 + c. Let l(j) = -118*e(j) + 54*u(j). What is the derivative of l(a)?`
Answer: `546*a**2 - 108*a - 118`
Question: `Three letters picked without replacement from qqqkkklkqkkk. Give prob of sequence qql.`
Answer: `1/110`

Figure 1: Examples from the dataset.

solving simple math problems: language processing ‘reinforcement learning’

‘language transformer models’

Solving Quantitative Reasoning Problems with Language Models

Aitor Lewkowycz*, Anders Andreassen†, David Dohan†, Ethan Dyer†, Henryk Michalewski†, Vinay Ramasesh†, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur*, Guy Gur-Ari*, and Vedant Misra*

Google Research

Abstract

Language models have achieved remarkable performance on a wide range of tasks that require natural language understanding. Nevertheless, state-of-the-art models have generally struggled with tasks that require quantitative reasoning, such as solving mathematics, science, and engineering problems at the college level. To help close this gap, we introduce Minerva , a large language model pretrained on general natural language data and further trained on technical content. The model achieves state-of-the-art performance on technical benchmarks without the use of external tools. We also evaluate our model on over two hundred undergraduate-level problems in physics, biology, chemistry, economics, and other sciences that require quantitative reasoning, and find that the model can correctly answer nearly a third of them.

https://minerva-demo.github.io

examples taken from
Matt Schwartz at Amplitudes 2022, Prague

Simplifying Polylogarithms with Machine Learning

Aurélien Dersy, Matthew D. Schwartz and Xiaoyuan Zhang

adersy@g.harvard.edu, schwartz@g.harvard.edu, xiaoyuanzhang@g.harvard.edu

Department of Physics, Harvard University, Cambridge MA 02138

The NSF AI Institute for Artificial Intelligence and Fundamental Interactions

Abstract

Polylogrithmic functions, such as the logarithm or dilogarithm, satisfy a number of algebraic identities. For the logarithm, all the identities follow from the product rule. For the dilogarithm and higher-weight classical polylogarithms, the identities can involve five functions or more. In many calculations relevant to particle physics, complicated combinations of polylogarithms often arise from Feynman integrals. Although the initial expressions resulting from the integration usually simplify, it is often difficult to know which identities to apply and in what order. To address this bottleneck, we explore to what extent machine learning methods can help. We consider both a reinforcement learning approach, where the identities are analogous to moves in a game, and a transformer network approach, where the problem is viewed analogously to a language-translation task. While both methods are effective, the transformer network appears more powerful and holds promise for practical use in symbolic manipulation tasks in mathematical physics.

$$\ln(x) = \int_0^x \frac{d_t}{t}$$

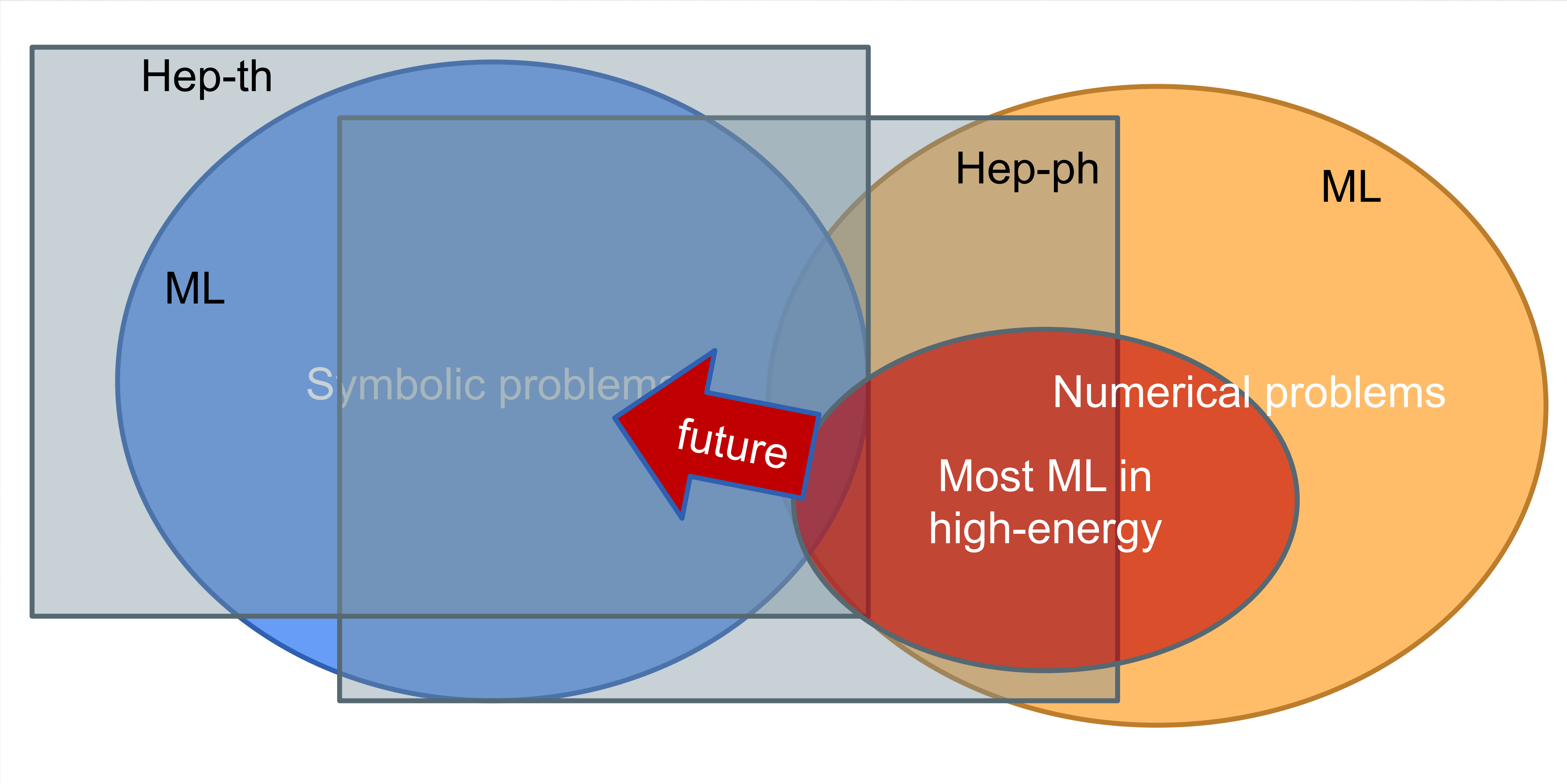
$$\mathrm{Li}_2(x) = \int_0^x \frac{dt_1}{t_1} \int_0^{t_1} \frac{dt_2}{1-t_2}$$

$$\ln(xy) = \ln x + \ln y$$

$$\mathrm{Li}_2(x) = -\mathrm{Li}_2(1-x) + \frac{\pi^2}{6} - \ln(x)\ln(1-x)$$

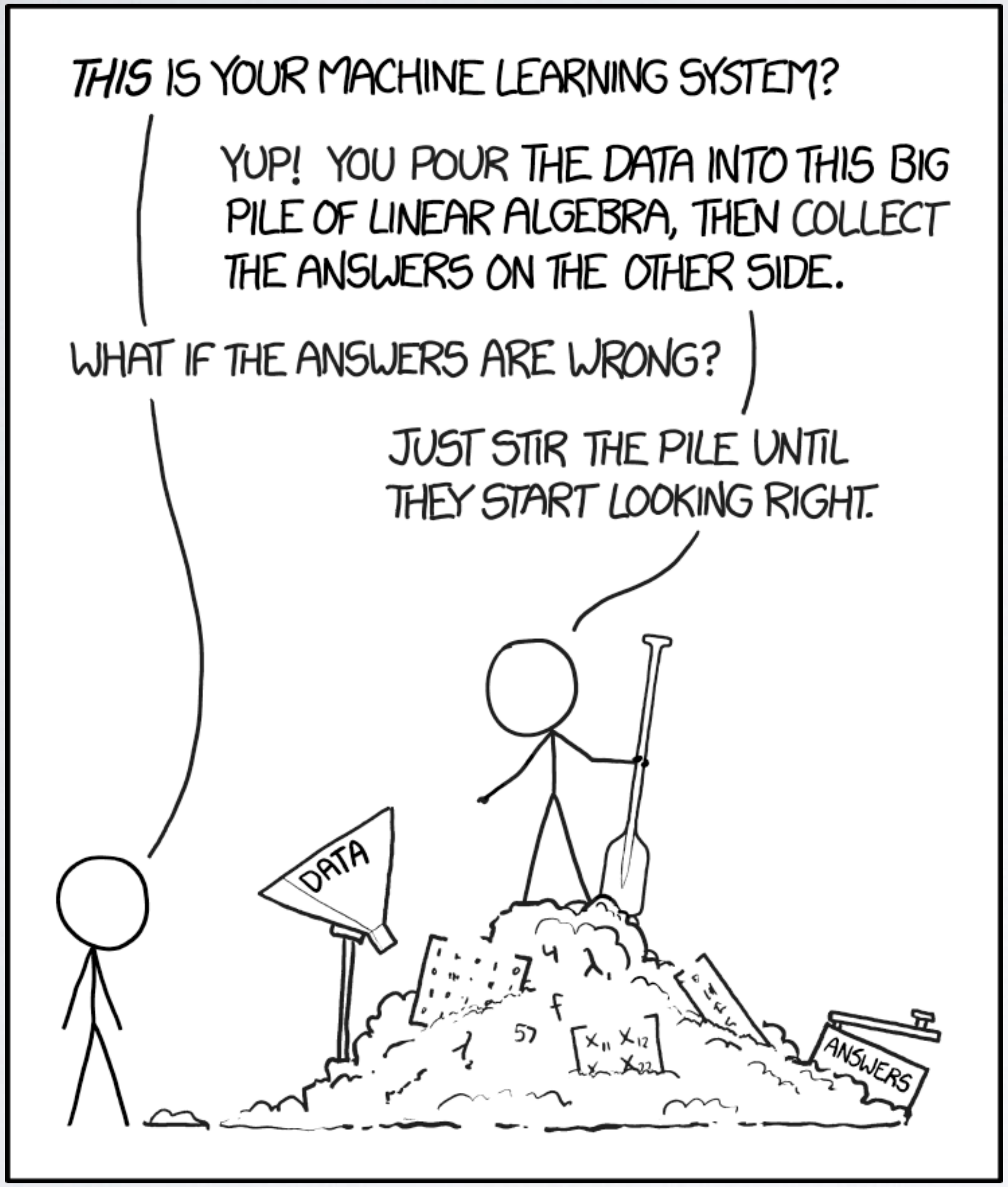
$$\mathrm{Li}_2(x) + \mathrm{Li}_2(y) + \mathrm{Li}_2\left(\frac{1-x}{1-xy}\right) + \mathrm{Li}_2(1-xy) + \mathrm{Li}_2\left(\frac{1-y}{1-xy}\right)$$
$$= \frac{\pi^2}{2} - \ln(x)\ln(1-x) - \ln(y)\ln(1-y) - \ln\left(\frac{1-x}{1-xy}\right)\ln\left(\frac{1-y}{1-xy}\right)$$

some are even convinced this is the future, maybe they are right?



Matt Schwartz at Amplitudes 2022, Prague

ML promises a lot, but we need to practice first…



https://xkcd.com/1838/