New Directions in Machine Learning Algorithms

Harrison B. Prosper

Florida State University

5th CMS Single Top Workshop

29 November, 2018

Topics

- Introduction
- New Directions
- Summary

INTRODUCTION

A Bit of History

- In 2009, after more than a *decade* of sustained effort, CDF and D0 reported the observation of single top quark production at the Fermilab Tevatron.

- It is, therefore, fitting to discuss machine learning in a single top workshop because the high-profile searches by CDF and D0 were the first at a hadron collider to make *extensive* use of multivariate methods.

- It was already clear in 1997 that had we stayed with traditional methods, the search would have taken much longer.

A Bit of History

- Three methods were used in these searches:
  - Boosted decision trees (BDT)
  - Bayesian neural networks (BNN)
  - Matrix element method (ME)

- BDTs are now widely used in ATLAS and CMS and, in at least one publication, the CMS Single Top Group made use of neural networks (NN) (*Physics Letters B doi:10.1016/j.physletb.2017.07.047*).

- The ME method, reincarnated as MELA, is widely used within the CMS Higgs to ZZ to 4-Lepton Group.

NEW DIRECTIONS

New Directions

In the following, I’ll describe three “new” directions that may be of interest to single top researchers:

❖ Going deep	Use of highly nested models
❖ Going raw	Use of low-level training data
❖ Going automatic	Use of automatic optimization of training parameters

Going Deep

- In 2006*, Hinton, Osindero, and Teh succeeded in training a multilayer neural network.

- This breakthrough re-ignited interest in very large, highly non-linear, models $f(\boldsymbol{x}, \theta)$, now generically referred to as *deep neural networks* (DNN).

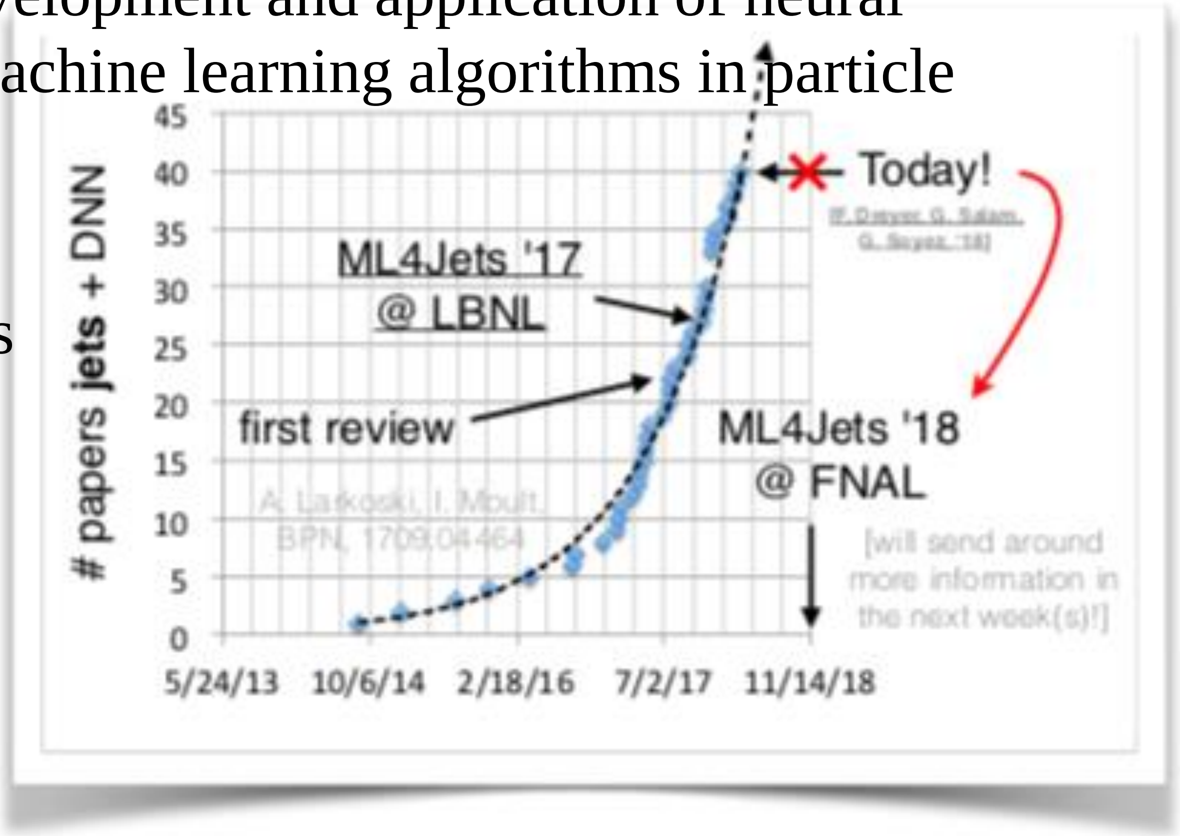
- Meanwhile, the use of BDTs became the machine learning method of choice within ATLAS and CMS.

* Hinton, G. E., Osindero, S. and Teh, Y. (HOT), A fast learning algorithm for deep belief nets, Neural Computation 18, 1527-1554.

Going Deep

- However, in the past few years, there has been significant growth in the development and application of neural network based machine learning algorithms in particle physics.

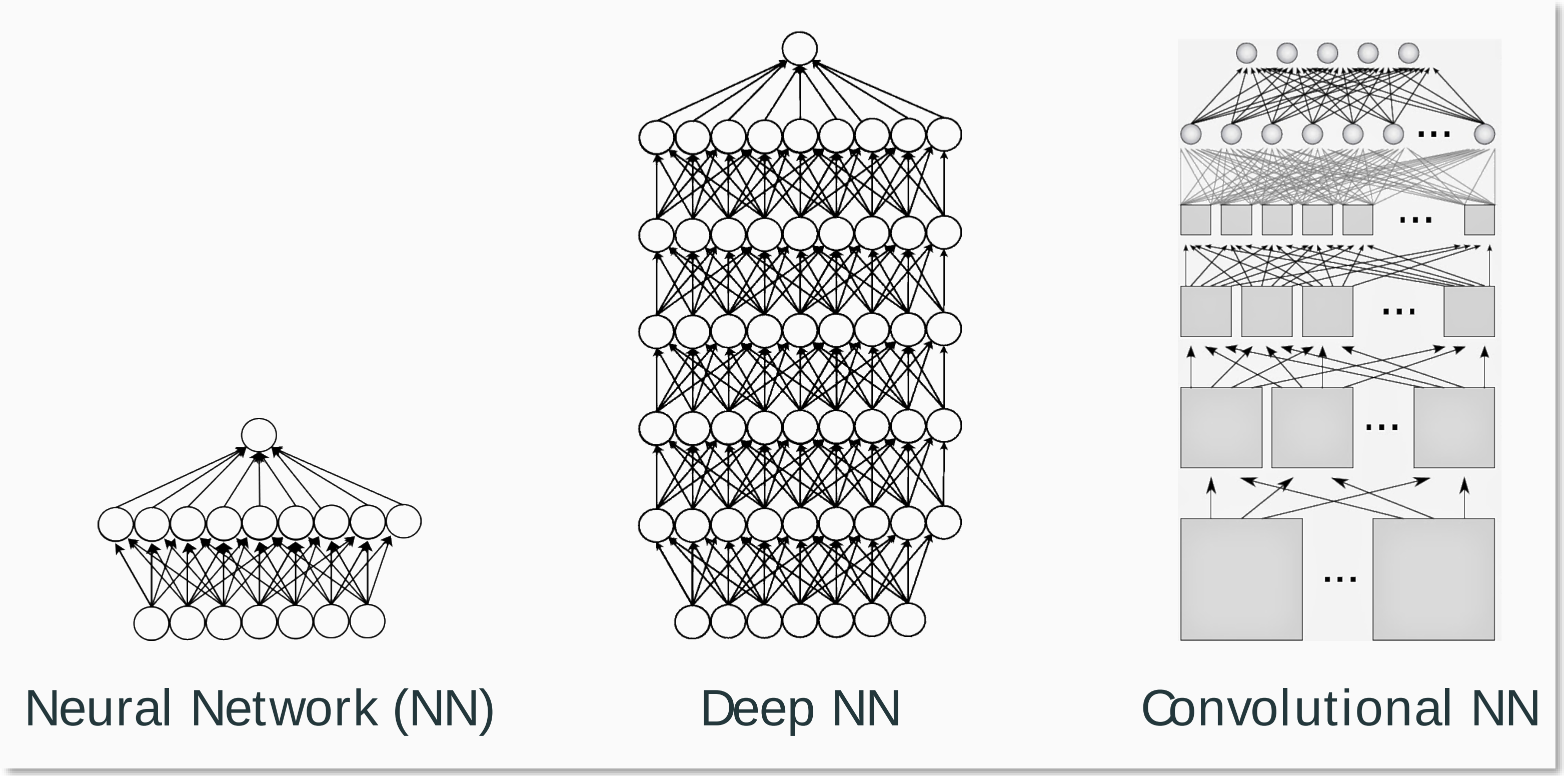
- This graph shows the number of papers on the application of ML to jets.



Sergei Gleyzer Fermilab Wine and Cheese Seminar, 11/16/18

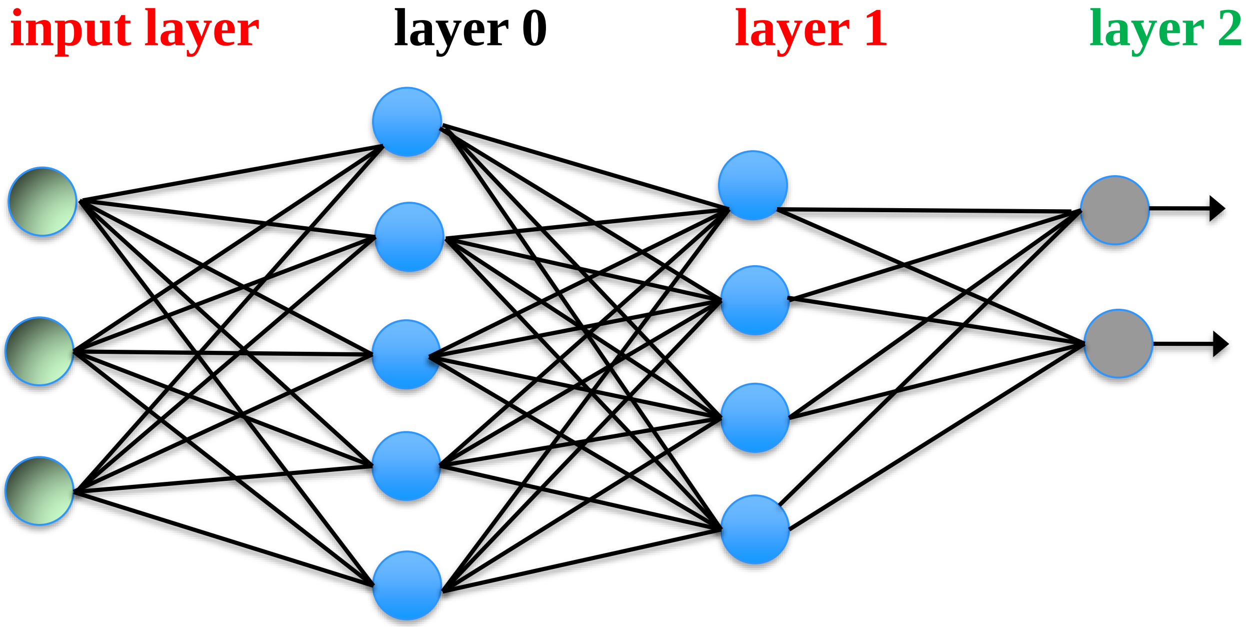
Going Deep: $f(\boldsymbol{x},\theta)$

In addition to going deep, a particular focus has been the exploration of different ways of connecting layers and of processing their outputs.



Sergei Gleyzer Fermilab Wine and Cheese Seminar, 11/16/18

Going Deep: Fully Connected DNNs



A 3-layer DNN

$$o = g(\boldsymbol{b}_2 + \boldsymbol{w}_2 h(\boldsymbol{b}_1 + \boldsymbol{w}_1 h(\boldsymbol{b}_0 + \boldsymbol{w}_0 x)))$$

h(z) = ReLU(z) [= max(0, z)], tanh(z)

g(z) = Identity(z), logistic(z) = 1/[1 + exp(-z)]

Going Deep: Fully Connected DNNs

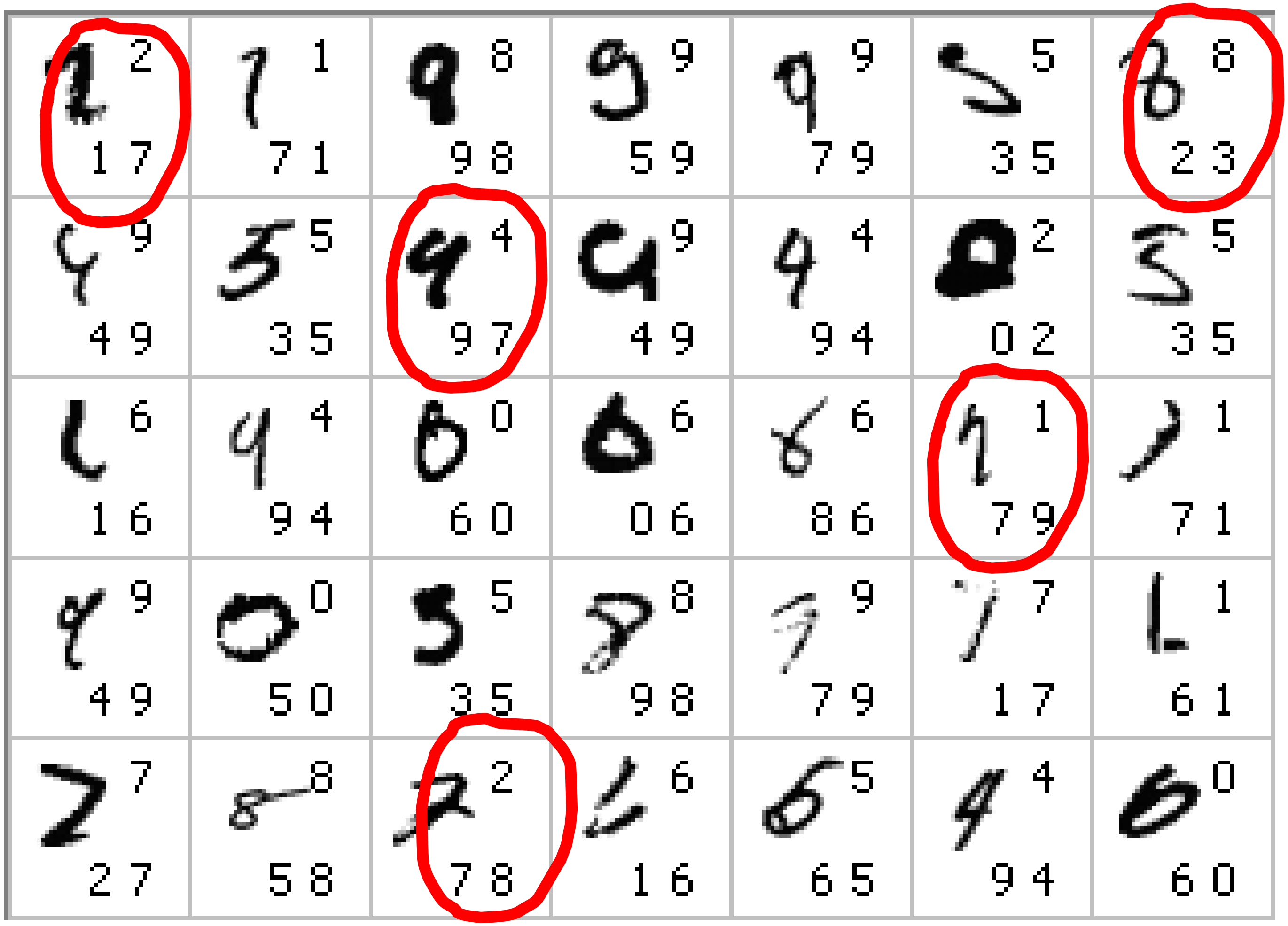
- ❖ In 2010, Cireşan *et al.** succeeded in training a DNN with architecture (**784**, 2500, 2000, 1500, 1000, 500, **10**) that classified the hand-written digits in the MNIST database.

- ❖ The database comprises 60,000 28 × 28 = 784 pixel images for training and validation, and 10,000 for testing.

- ❖ The error rate of their ~12-million parameter DNN was 35 images out of 10,000. The misclassified images are shown on the next slide.

* Cireşan DC, Meier U, Gambardella LM, Schmidhuber J. , Deep, big, simple neural nets for handwritten digit recognition. Neural Comput. 2010 Dec; 22 (12): 3207-20. http://yann.lecun.com/exdb/mnist/

(784, 2500, 2000, 1500, 1000, 500, 10)



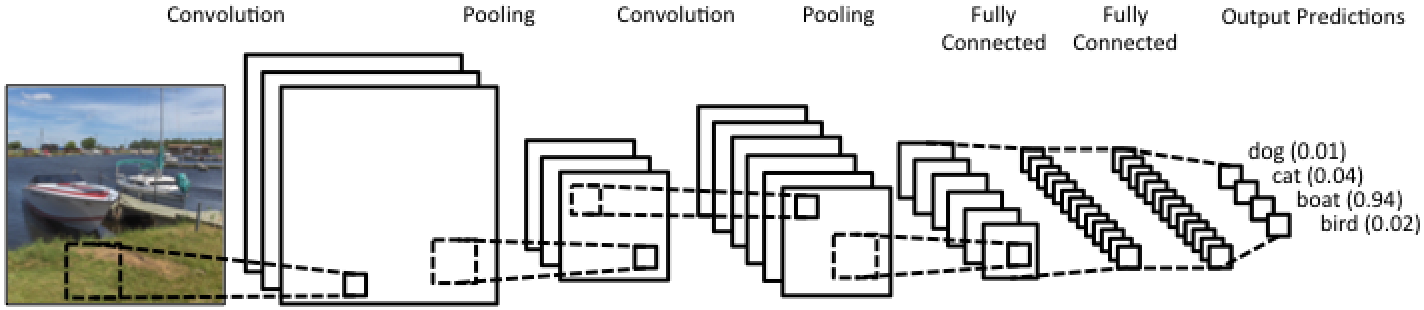
Upper right: correct answer; lower left answer of highest DNN output; lower right answer of next highest DNN output.

Going Deep, Sparse, and Raw

- Remarkable breakthroughs have occurred in image analysis and in playing games, such as Go, which make use of a class of deep neural networks (DNN) called convolutional neural networks (CNN).

- CNNs are *functions* that first compress data and then classify objects using their compressed representations.

Sou



Going Deep, Sparse, and Raw

Several groups have been exploring the use of deep, sparse models that make use of low-level data.

1. Removing unwanted particles in jets (Komiske *et al*., arXiv:1707.08600)
2. Quark/gluon jet classification (ATLAS, CMS)
3. End-to-end event classification (Andrews *et al*. axXiv:1807.11916v1)

In these applications *events*, or *event regions*, are treated as *images*. The inspiration arises from the advances in automated image recognition.

Deep, Sparse, Raw: Pileup Mitigation

LHC Run	Extra interactions per bunch crossing
Run 2:	20
Run 3:	80
Run 4:	200

Several approaches:

1. **P**ile**U**p **P**er **P**article **I**dentification (PUPPI)
   Bertolini, Harris, Low, and Tran, arXiv:1407.6013
2. Jet Cleansing
   Krohn, Low, Schwartz, Wang, arXiv:1309.4777
3. SoftKiller Cacciari, Salam, Soyez, arXiv:1407.0408
4. **P**ileup **M**itigation with **M**achine **L**earning (**PUMML**)
   Metodiev, Komiske, Nachman, Schwarz, arXiv:1707.08600

Deep, Sparse, Raw: PUMML

Basic idea

Treat a jet as a 3-color image in the (y, φ)-plane, where each color corresponds to be different category of particle.

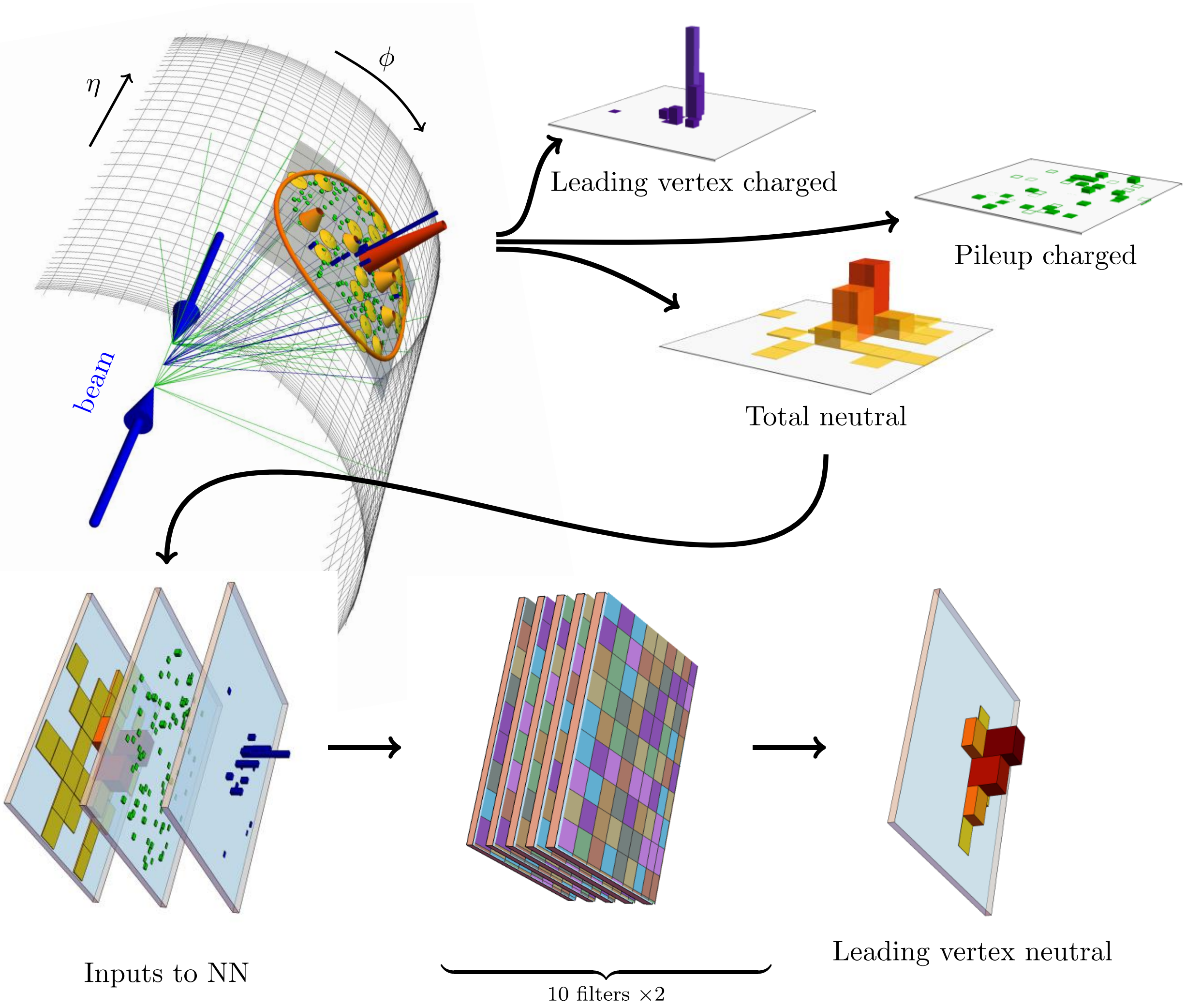
1. Red p_T of *all* neutral particles
2. Green p_T of charged pileup (PU) particles
3. Blue p_T of charged leading (primary) vertex (LV) particles

Construct a convolutional neural network to map the 3-color image to an image of the p_T of LV neutral particles.

The cleaned jet is then formed from the charged and neutral particles from the primary vertex.

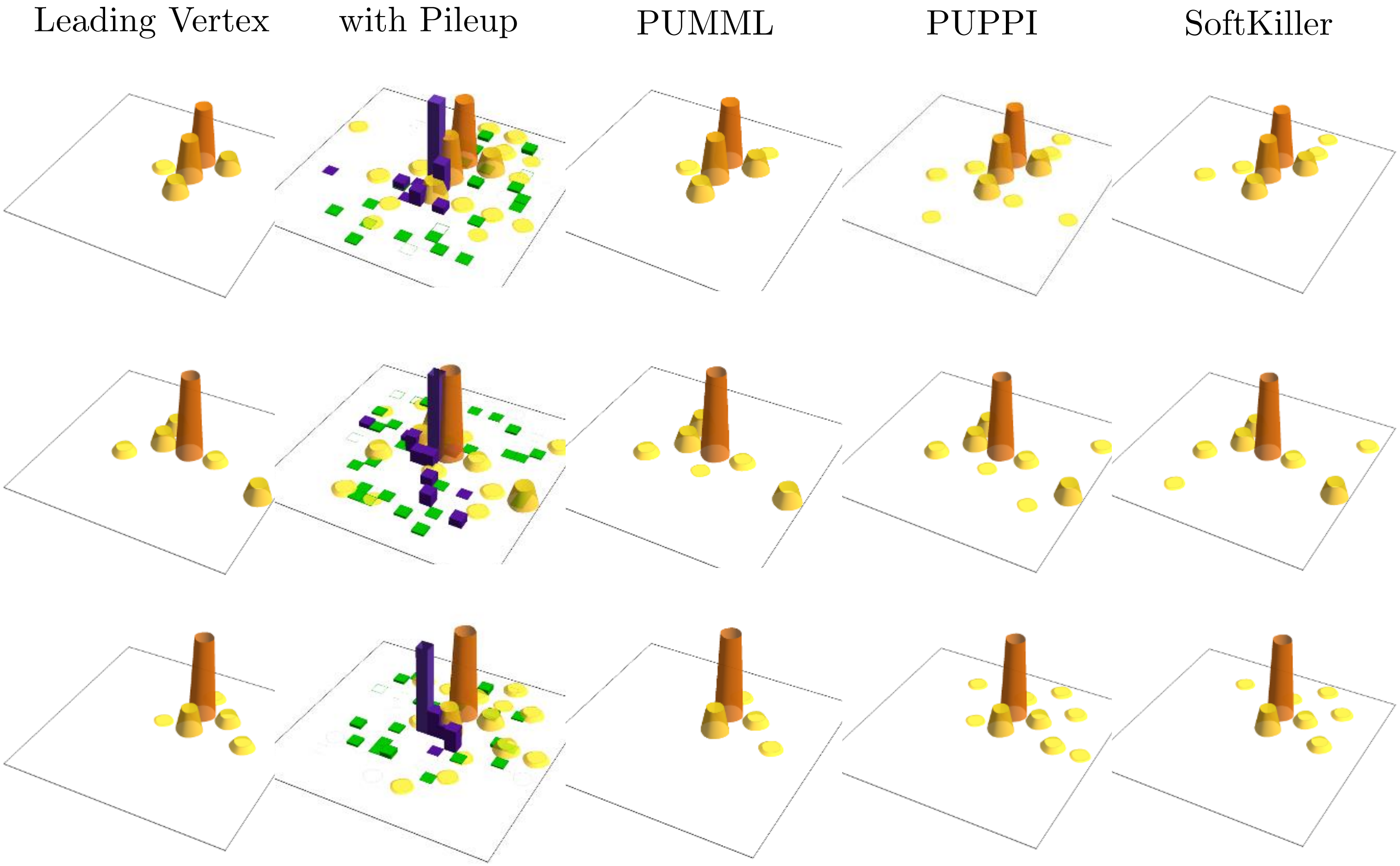
Metodiev, Komiske, Nachman, Schwarz, arXiv:1707.08600

Deep, Sparse, Raw: PUMML



JHEP 12 (2017) 051

Deep, Sparse, Raw: PUMML



JHEP 12 (2017) 051

Going Automatic

- When we optimize our analyses, ideally, we should maximize (or minimize) the quantity we really care about.

- For example, suppose we wish to measure the single top production cross section σ as accurately as possible, then we may wish to maximize the objective function

$$f(\alpha) = \sigma/\delta\sigma,$$

  which, in general, depends on parameters α for which choices have to be made in order to perform the analysis.

- Typically, $f(\alpha)$ is optimized by hand, say by PhD students and junior postdocs. Can this be automated?

Going Automatic: Bayes Optimization

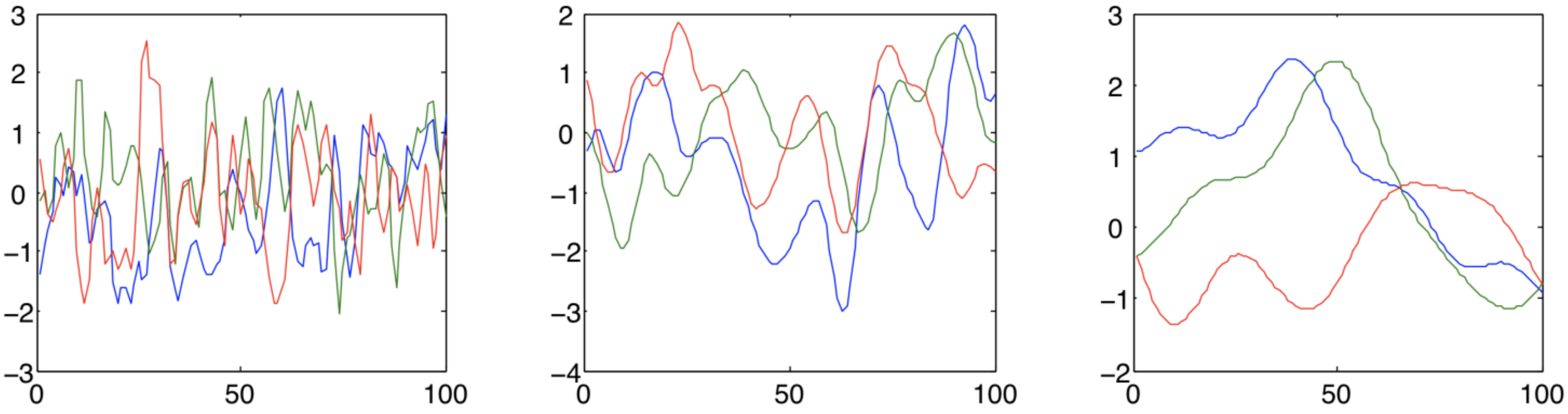
- While we don’t know the form of $f(\alpha)$, we can in principle compute it for any reasonable value of α.

- Bayesian optimization:
  1. Start with $f(\alpha)$ evaluated a n points so that $f_i = f(\alpha_i), i = 1, \ldots, n$ are known.
  2. Compute $p(f|F_n) = p(F_n|f)\pi(f)/p(F_n)$, where $F_n = f_i, \ldots, f_n$ and $F_n^* = \max(f_i, \ldots, f_n)$, and $\pi(f)$ is a prior density on some *function space*.
  3. Compute $A(\alpha) = \int [f(\alpha) - F_n^*]^+ p(f|F_n) df$
  4. Maximize $A(\alpha)$ with respect to α and call it α_{n+1}.
  5. Compute $f_{n+1} = f(\alpha_{n+1})$. Repeat 1…5 N times.

Going Automatic: Bayes Optimization

- In practice, Bayesian optimization uses a collection of function classes called a Gaussian process.

- Here is Fig. 2 from an excellent recent tutorial by Frazier*, which shows some functions $f(\alpha)$ (where $\dim \alpha = 1$) randomly drawn from three different Gaussian processes.

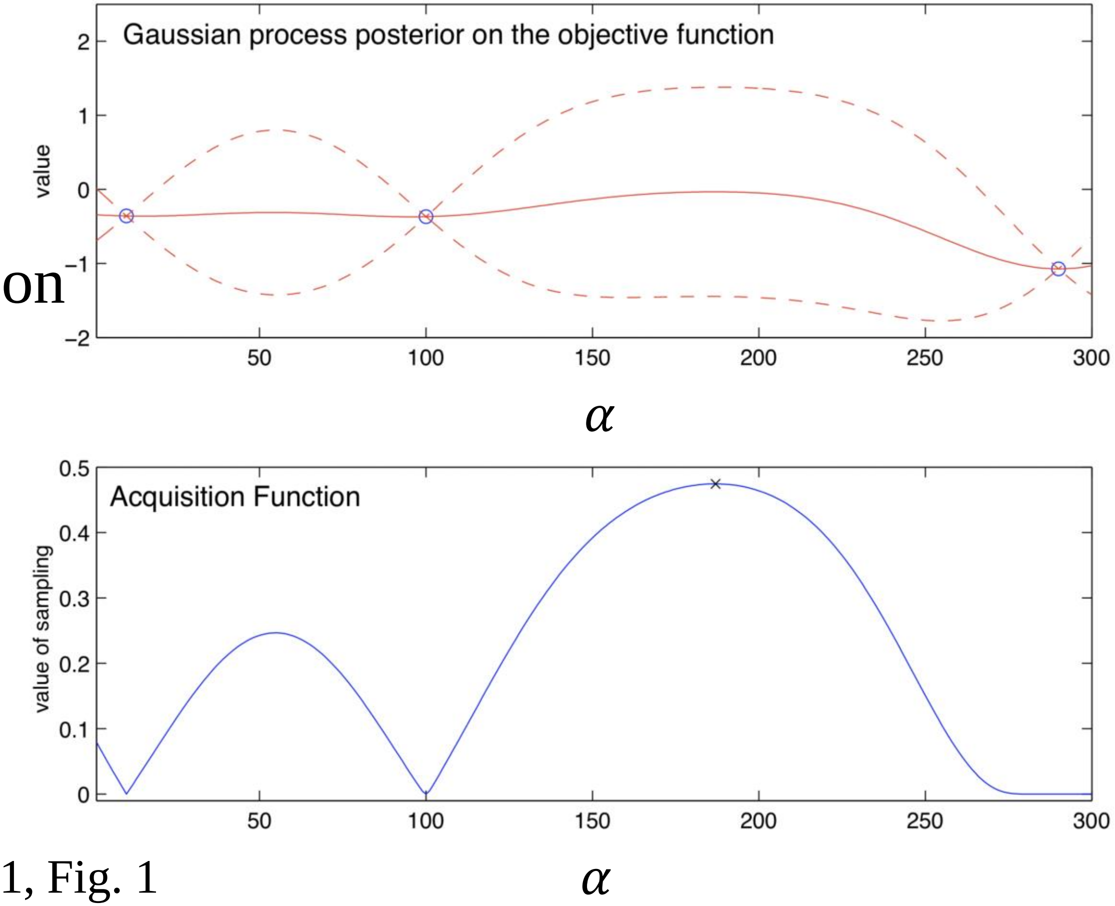


*Peter I. Frazier, arXiv:1807.0281v1, Fig. 2

Going Automatic: Bayes Optimization

- Here is Fig. 1 from the same paper, which (again for a 1-D function) shows three starting points $\alpha_1, \alpha_2,$ and α_3 and the associated values $f_1, f_2,$ and f_3.

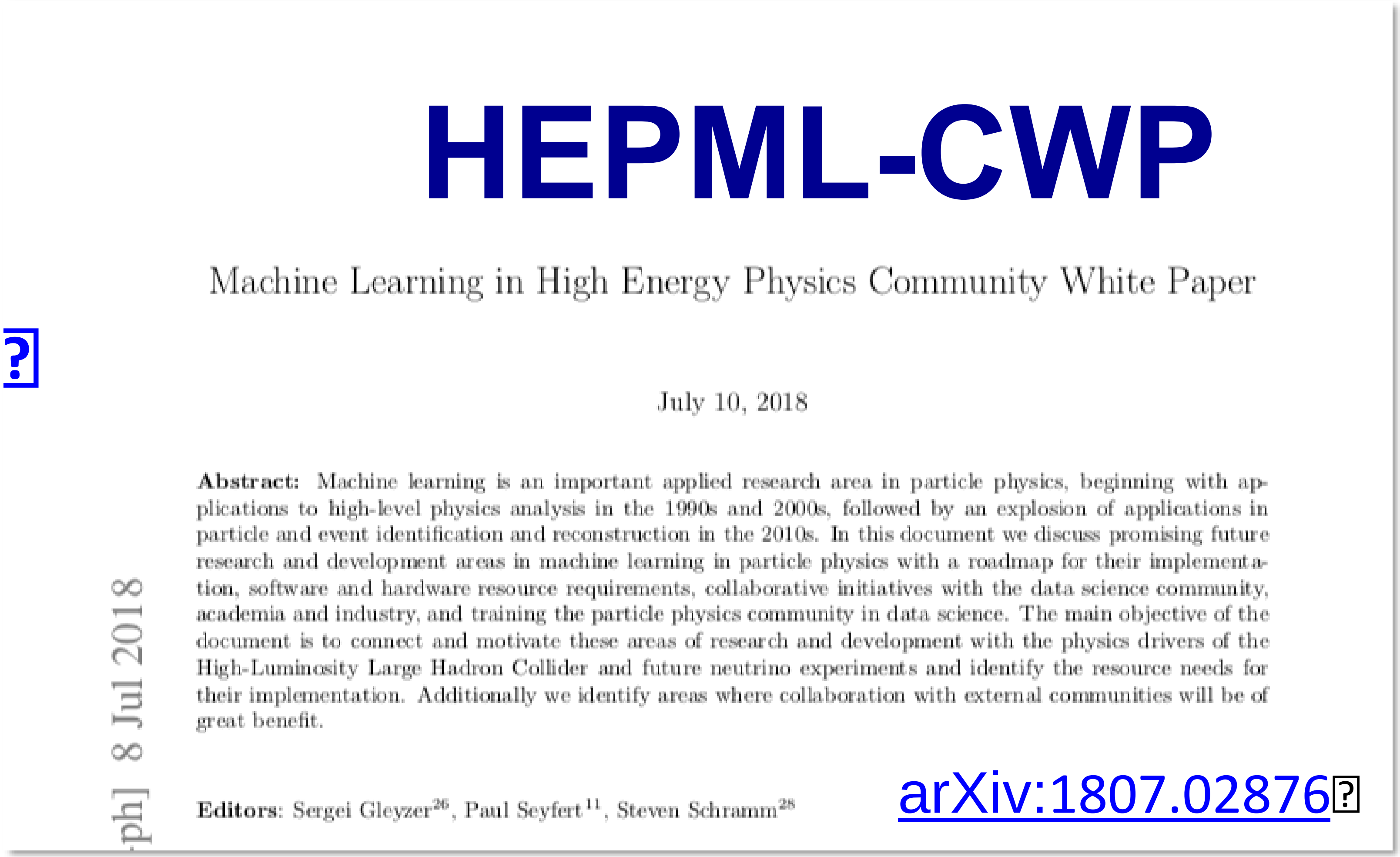
- The basic assumption of the Gaussian process is that the probability density of $f_i, \ldots, f_n$ is a multivariate Gaussian.



Peter I. Frazier, arXiv:1807.0281v1, Fig. 1

HEPML Community White Paper

?



HEPML-CWP

Machine Learning in High Energy Physics Community White Paper

July 10, 2018

Abstract: Machine learning is an important applied research area in particle physics, beginning with applications to high-level physics analysis in the 1990s and 2000s, followed by an explosion of applications in particle and event identification and reconstruction in the 2010s. In this document we discuss promising future research and development areas in machine learning in particle physics with a roadmap for their implementation, software and hardware resource requirements, collaborative initiatives with the data science community, academia and industry, and training the particle physics community in data science. The main objective of the document is to connect and motivate these areas of research and development with the physics drivers of the High-Luminosity Large Hadron Collider and future neutrino experiments and identify the resource needs for their implementation. Additionally we identify areas where collaboration with external communities will be of great benefit.

Editors: Sergei Gleyzer[26], Paul Seyfert[11], Steven Schramm[28]

arXiv:1807.02876

Summary

- While BDTs are the method of choice in our field, several particle physics groups are actively investigating the utility of deep neural networks for particle physics.

- There is evidence that bypassing reconstruction and using low-level data yields results comparable to those obtained with 4-vectors when the latter are accurately reconstructed and *superior* results when they are not.

- The results are very promising and should, at the very least, be looked at seriously.

BACKUP SLIDES

Machine Learning Algorithms

What is the goal?

One goal is to improve the discrimination between the single top quark signals and the background processes by using a multivariate function, $D(\boldsymbol{x})$, where $\boldsymbol{x}$ are discriminating variables (*features* in ML jargon).

How is it realized?

- Choose $\boldsymbol{x}$.
- Choose a model $f(\boldsymbol{x}, \theta)$.
- Choose a loss function $L(\boldsymbol{y}, f)$, where $\boldsymbol{y}$ is a target associated with $\boldsymbol{x}$ and $T = \{(\boldsymbol{y}_i, \boldsymbol{x}_i)\}$ are the training data.
- Choose a constraint $C(\theta)$ on the parameters.
- Minimize $R_N = \frac{1}{N}\sum_{i=1}^{N} L(\boldsymbol{y}_i, f(\boldsymbol{x}_i, \theta)) + C(\theta)$

Machine Learning Algorithms

Points to note:

- The function R_N (the empirical risk) approximates

$$R[f] = \int L\big(\boldsymbol{y}, f(\boldsymbol{x}, \theta)\big)\, p(\boldsymbol{y}, \boldsymbol{x}) d\boldsymbol{y} d\boldsymbol{x}$$

  where $p(\boldsymbol{y}, \boldsymbol{x})$ is the joint probability density of $\boldsymbol{y}$ and $\boldsymbol{x}$.

- The mathematical form of the function $D(\boldsymbol{x})$ approximated by the model $f(\boldsymbol{x}, \theta)$ is determined by the form of the loss function. In particular, it is independent of the details of the model $f(\boldsymbol{x}, \theta)$ provided that
  1. the model is sufficiently flexible, and
  2. the training data T are sufficiently numerous.