Lecture III : ML in HEP

Outline :

(i) Overview of applications

(ii) my own research : Bayesian neural networks

Overview

→ This is not a complete list !

→ Check out the great collection of papers about modern ML : HEPML: A Living Review !

---

improving an analysis

→ Event selection , tagging

↳ b - tagging

↳ top tagging

↳ light quarks vs. gluons

↳ full events

⋮

BDT's and shallow NN already used for decades !

for some deep learning review : arxiv: 1902.09914

→ object reconstruction / identification / calibration

⇒ Countless number of papers from ATLAS / CMS / LHCb , ...

---

model independent searches

→ Anomaly detection , new physics searches , use of Auto encoders

⇒ HEPML → section about anomaly detection

→ Unfolding : → OmniFold (arxiv: 1911.09107)

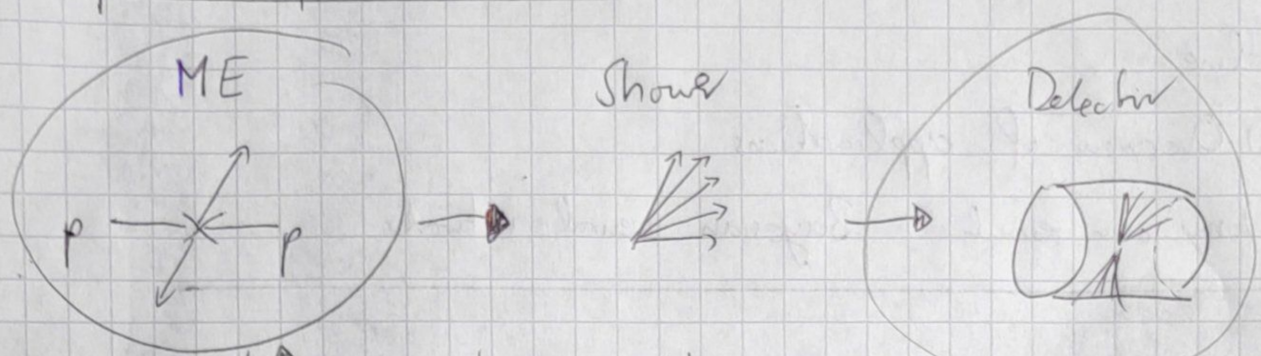
→ arxiv: 1912.00477 (Bellagente et al.)

⋮

(HEPML → Unfolding)

→ PileUp removal (PUMML, ... → HEPML → Regression → PileUp)

→ Speed up simulations:



↳ Expensive! Replace by NN!

arxiv: 1907.03764
and many more!
→ HEPML → section about GAN or Flows!

→ CaloFlow / CaloGAN (arxiv: 1712.10321)

and more!

→ NNPDF (PDF description via NN)

→ String theory landscape and Deep Learning (→ J. Halverson)

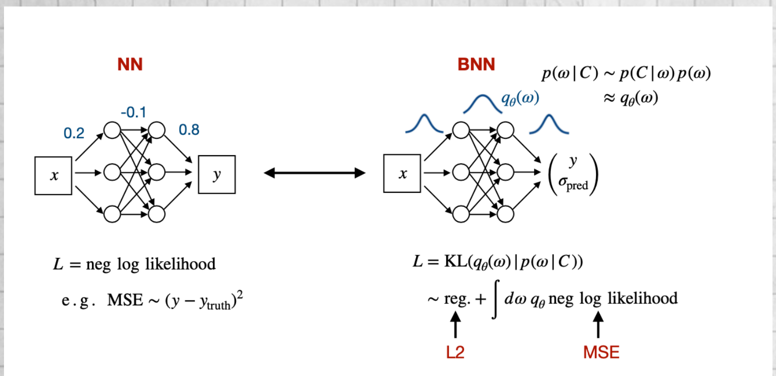
→ Solving diff. equations with neural networks
↳ Cosmology: arxiv: 1902.05563)

⋮

Bayesian neural networks (Sketch of my own research)

Motivation: Add uncertainties for Deep Learning

What is a BNN?



Variational inference:

$$p(\omega|D) \sim p(D|\omega)\; p(\omega)$$
$$\approx G(\omega\,|\,\underbrace{\mu_\omega, \sigma_\omega}_{\text{trainable parameters!}})$$

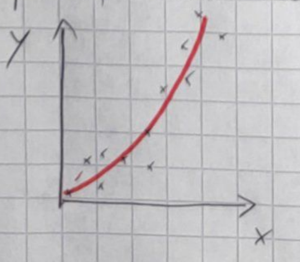
→ minimize difference between $p(\omega|D)$ and $G(\omega|\mu_\omega, \sigma_\omega)$!

We applied method to:

- → top tagging
- → regression/calibration: top pt reconstruction   here
- → generative task: (simulation of ME) (Bayesian invertible neural net)

Regression with uncertainties

Example of second lecture



Loss: MSE

$$|y - \underset{\uparrow\ \text{NN}}{f(x,\omega)}|^2$$

how to ~~add~~ "teach" our network to output an uncertainty?

→ Replace MSE with full Gaussian neg log likelihood!

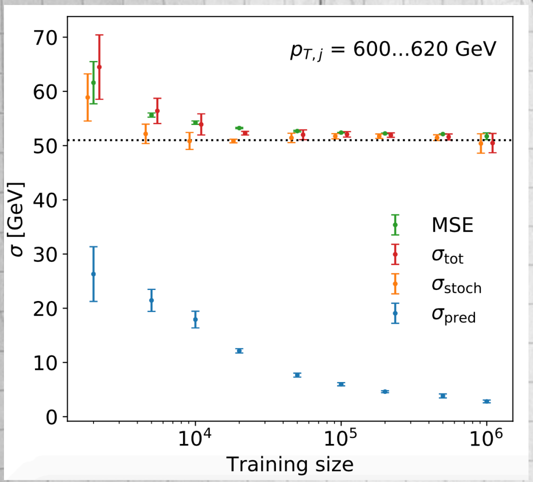
$$\frac{|y - f(x,\omega)|^2}{2\sigma^2} + \frac{1}{2}\log\sigma^2$$

→ ~~a~~ make σ an output of the NN!

$$x \longrightarrow NN \begin{cases} y(x) \\ \sigma(x) \end{cases}$$

→ this doesn't capture the fact that our NN is imperfect! Finite training size!

$$\Rightarrow BNN \begin{cases} y(x) \\ \sigma(x) \\ \sigma_{pred}(x) \end{cases}$$ (from weight unc)



→ σ_{pred} is training size dependent

→ ~~σ_{stoch}~~ σ_{tot} coincide with MSE (error of network)