ML: a few practical considerations

Huilin Qu (曲慧麟)

Weekly Meeting of IHEP ML Innovation group

2023-02-28

CERN

DISCLAIMER

- Based on my very personal experiences in using ML to solved HEP problems
  - and highly biased to jet tagging
  - so, please take them with a large grain of salt

- ML is 50% mathematics and 50% engineering
  - and probably another 50% alchemy...

Data Matters

- Always inspect your training data
  - check the distributions for different classes / in different phase space (p_T, energy scale, vs time, …)
    - do they make sense?
    - are the trends expected?
    - do you see expected / unexpected separation power between different classes?
  - check for significant outliers / NaN / Inf / etc.

- Think carefully about how to choose your training data, how to define training target (truth labels, etc.)
  - highly case dependent, but this can have significant impact on the performance, generalization power, etc.

Data Matters (II)

- Mindful preprocessing
  - NNs work best with Gaussian-like inputs
    - transform the inputs if needed, e.g., log(...) or tanh(c ...) for long-tail distributions (energy, p_T, mass, d_0/dz, ...)
    - shift/scale the inputs, and then truncate (if needed) – extreme outliers can destabilize training and affect performance
    - use normalization layers (BatchNorm, LayerNorm, ...)
  - dealing with phase space difference between classes => reweighting (or better, sampling) if needed
  - decorrelation (e.g., mass decorrelation in jet tagging) – see e.g., link

- Get more data whenever you can
  - if can not: consider data augmentation (rotation, reflection, smearing, ...)

Baseline First, then Iterate

- A good practice is to always establish a baseline algorithm first before developing more advanced approaches
  - with a baseline ready, then one can easily evaluate if the new algorithm is too good (to be true), or too bad (so probably missing something obvious), or just promising :)
  - if the problem is not new and a baseline already exists – just use it
  - the baseline can be a cut-based / rule-based algo, or a shallow model (e.g., BDT)
    - consider trying newer BDT libraries like TensorFlow Decision Forests (TF-DF), LightGBM, CatBoost – in addition to xgboost
    -

Some Useful Links

- Tutorial / hands-on textbook:
  - Dive into Deep Learning / 《动手学深度学习》: https://d2l.ai/

- A Living Review of Machine Learning for Particle Physics
  - https://github.com/iml-wg/HEPML-LivingReview

- My little framework:
  - weaver-core: https://github.com/hqucms/weaver-core
  - weaver-examples (under construction): https://github.com/hqucms/weaver-examples