Second Assignment 2021

Machine Learning in Particle Physics and Astronomy

Start date: March 1, 2021
Deadline: April 28, 2021

Currently a hot topic at the Large Hadron Collider is the search for the production of 4 of the heaviest quark (the top quark) in a single event. These special events are called "4-top events". A signal for the production of these events has not been observed so far. The discovery of "4-top events" is interesting since it might be that more events are measured than expected. This could point to physics beyond the Standard Model of particle physics, since new physical processes could also generate events with 4 top quarks. The top quarks decay to other particles which are then measured in the detector surrounding the point where the quarks were originally produced.

The simulated training and validation data are provided in a one-line-per-event text format (CSV), where each line has variable length and contains 5 event-level quantities followed by low-level features for each object in the event. The format of CSV files are (in one line):

```
event ID; process ID; event weight; MET; METphi; obj1, E1, pt1, eta1, phi1; obj2, E2,
                                    pt2, eta2, phi2;  ...
```

`obj` specifies the particles detected in the event (e.g. electron (e), photon (p), a so-called jet (j), a so-called b-jet (b) a muon (m) etc.). The + or - specifies the charge of the particle. The `E,pt,eta,phi` specify the 4-vector of the measured particle, i.e. the energy, transverse component of the momentum and the theta (given here in units of pseudorapidity) and phi angles. The `event ID` is a serial integer to uniquely identify that particular event in the run. The `event weight` is a real number to determine the likelihood of the process. This is a generator quantity and does not exist in real data. It should not be used for "training". The `process ID` is a string referring to the process which generated the event. In real life events this is unknown on a event-by-event basis, but in simulated data sets (like the one we are providing you with) this information *is* accessible. The `MET` and `METphi` entries are the magnitude and the azimuthal angle of the missing transverse energy vector of the event, respectively. "Missing" means that this momentum is taken away by undetected particles, e.g. neutrinos.

As an example, an event corresponding to the final state of the background $t\bar{t} + 2j$ process with two b-jets and one jet reads as follows:

94;ttbar;0.00167779;112288;1.74766;b,331927,147558,-1.44969,-1.76399;j,100406,85589,-0.568259,-1.17144;b,55808.8,54391.4,-0.198215,1.726

1 Assignment

We have compiled a dataset based on simulation of events produced at the Large Hadron Collider which you can use to train and validate your model. This dataset can be found on Brightspace. In a few weeks we will provide a second datasets (test data) without labels, please use your algorithm to predict the process. This second datasets also comes with priors for the different processes.

Your task is to create **three models** that are able to distinguish background from 4-top candidate signal events. The three models you have to create are:

a) **a simple discriminator neural network** that can categorize events on an event-by-event basis into signal (4top) and background (all other categories).

Use the event information as inputs (but do **not** use the process ID, event ID or event weight). The problem here is also that the size of the information changes for each event. Fill missing information with 0 (zero padding).

b) a **multiclass discriminator neural network** that can categorize events on an event-by-event basis into the categories 4top, ttbarHiggs, ttbarW, ttbarZ and ttbar.

Take into account Bayesian formula to derive from the classification output posterior probabilities for each class depending on the priors you assume.

c) Propose a way to use the different class probabilities (4top, ttbarW, ttbarHiggs, ttbarZ, ttbar) to optimally distinguish 4top events from backgrounds. Apply this method to the training/validation data. Does it perform better than the simple discriminator network in exercise a ?

d) Try to improve the network architecture to optimally distinguish signal (4top) from the background processes. You can try e.g. convolutions or recurrence or graph representations of your data. Your creativity has no boundaries here.

2 Handing in

The deadline for the assignment is printed at the top of the first page. Any assignment handed-in after this deadline will not be graded. Handing in your assignment is done through Brightspace. You need to hand in **four** elements. Not handing in all four before the deadline will result in us not grading your solution.

Element 1: A scientific paper describing your solution

Using your development logbook you use during the development of the algorithm, create a scientific paper about your solution. This write-up should be in the form of a paper of maximum 4 pages long letter like the ones published by Physical Review Letters (PRL) as a reference for this[1]. Use the following structure:

- Start with an abstract;
- Describe the problem;
- Describe the investigated methods and why you used these;
- Report and discuss your results;
- Conclude with a conclusion;
- Add an appendix with technicalities[2].

A seminar on how to structure a scientific paper will be given on the 18th March.

Element 2: Your code

Your code in .py or .ipynb format.

Element 3: The trained algorithms

The models you trained in saved format. What kind of file you hand in, depends on the used library:

- **keras**: Use the `.save` method of the keras model to save it to an .hdf5 file. See `https://keras.io/getting-started/faq/#how-can-i-save-a-keras-model`.
- **sklearn**: Use the built in **joblib** package in python to store the object containing your trained model. See `https://scikit-learn.org/stable/modules/model_persistence.html`.

Please also include a script to read and run your saved model(s).

Element 4: Predictions by your algorithms

For each of your algorithms, create a file with its predictions on the testing data provided on Brightspace. Put each prediction on its own line, prepended by the ID of the event the prediction belongs so. Separate the ID and the prediction by a comma. three lines of your file might therefore look like this:

```
121561,ttbar
121562,ttbar
121562,4top
```

This information will be used to grade the performance of your algorithm.

[1] For example: `https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.122.014502`.
[2] PRL normally does not allow for appendices, but for the purpose of this assignment we allow you to add one anyway.

3 Grading

The final grade will be constructed based on a grading of the paper and the code. The write-up (and performance) contributes 70% of the final grade, whereas the code contributes 30%. The write-up will be checked on the following:

- Does it show the student understands the theoretical concepts of the applied methods?
- Does the choice for applied (machine learning) methods make sense?
- Is the student concise in argumentation?
- Given the argumentation and available hardware, did the student find an algorithm that suits the problem?
- Was any optimisation performed and, if so, was this optimisation done in a sound manner?
- Is the write-up written like a scientific paper?

The code will be graded based on the following criteria:

- Does the code do what the student described in the paper?
- Is the implementation of the solutions correct? (e.g. does it *run*?)
- Do the algorithms work and perform as described?
- Is the code readable (e.g. does it contain enough explanation in the form of comments for someone who has never read the code to understand it? it is structured logically?)