documentation nn_unwgt

lucabeccatini

October 11, 2022

1 Settings

The settings representing some features of the programs are:

- "seed_all": seed for tf and np.
- "norm": it defines the inputs and the their normalization. "s_pro" and "s_int" use as inputs 6 momenta normalized respectively respect the E_{cm} of proton and the E_{cm} of the interaction. "cmass" uses 5 inputs which are: the ratio between the E_{cm} of proton and the E_{cm} of the interaction, the rapidity of the top in the laboratory frame, the 3 momenta of the top in the center of mass frame.
- "output": the output of the nn. With "wgt" the nn predicts the weights of the events, with "lnw" it predicts the natural logarithm of them.
- "lossfunc": the loss function used in the training. "mse" is the mean squared error, "chi" is the chi square function.
- "maxfunc": maxima function used for the unweighting. "mqr" is the maximum quantile reduction, "mmr" is the maximum median reduction.
- "unwgt": unweighting method. "new" is the new method where the first unweighting is performed respect s_max and the final weight is witten as $s = max(max())$. "pap" is the method of the paper where the first unweighting is performed respect w_max and the final weight is written as $s = max()max()$.

The constants and functions defined at the beginning are:

- "eff1_st": efficiencies of MG5 during the first unweighting for the same samples of events. It is around 10% because MG uses a maximum about 10 times larger then the real one to do the first unweighting.
- "eff2_st": efficiencies of MG5 during the second unweighting for the same samples of events. At least for simple cases, it is about 99% because MG excludes only up to 5 events in the second unweighting (these events, up to 5, are not unweighted by MG in the first unweighting, probably for case with small events).

- "f_kish": Kish factor, defined such that the Kish effective sample size N_{eff} is

$$N_{eff} = \frac{(\sum_j^N w_j)^2}{\sum_j^N w_j^2} = \alpha N \tag{1}$$

thus the Kish factor is: $\alpha = \frac{(\sum_j^N w_j)^2}{N \sum_j^N w_j^2}$. α is always lower than 1 if there are overweighted events and and equal to 1 if there aren't.

- "efficiency": efficiency of the unweighting defined as

$$eff = \frac{\alpha_f N_f}{N_i} = \frac{(\sum_j^{N_f} z_j)^2}{N_f \sum_j^{N_f} z_j^2} \frac{\sum_j^{N_i} s_j}{N_i s_{max}} \tag{2}$$

where N_i and s_j are respectively the number of events and their weights before the unweighting, N_f and z_j are the number of events and their weights after the unweighting, s_{max} is the constant respect which the unweighting is performed.

- "effective_gain": effective gain factor between the standard method and the new one

$$f\_eff = \frac{N \frac{t_{st}}{\epsilon_{st}}}{\frac{N}{\alpha}(\frac{t_{nn}}{\epsilon_1 \epsilon_2} + \frac{t_{st}}{\epsilon_2})} = \frac{\alpha}{\frac{t_{nn}}{t_{st}} \frac{\epsilon_{st}}{\epsilon_1 \epsilon_2} + \frac{\epsilon_{st}}{\epsilon_2}} \tag{3}$$

where N is the number of produced events, $\frac{t_{st}}{t_{nn}}$ is the ratio between the mean time of the standard method to compute one event and the one of the new method, $\epsilon_{st} = eff1\_st \times eff2\_st$ is the standard efficiency, $\epsilon_1 = eff\_1$ and $\epsilon_2 = eff\_2$ are respectively the efficiency of the first and second step of the new method. α is the total Kish factor of the unweighting.

2 Prediction of weights

The momenta and weights of events are read from "info_wgt_events_5iter.txt", where in each line there are: $p_x^t, p_y^t, p_z^t, E^t, p_z^{\bar{t}}, E^{\bar{t}}, weight$ of a given event.

The energy of the center of mass of the interaction is defined as

$$E\_cm\_int = \sqrt{(E^t + E^{\bar{t}})^2 - (p_x^t + p_x^{\bar{t}})^2 - (p_y^t + p_y^{\bar{t}})^2 - (p_z^t + p_z^{\bar{t}})^2} \quad . \tag{4}$$

The beta function of the top-antitop pair in the lab frame is:

$$\beta = \sqrt{\frac{|p_z^t + p_z^{\bar{t}}|}{E^t + E^{\bar{t}}}} \quad . \tag{5}$$

The rapidity of the top in the lab frame is defined as

$$y = \frac{1}{2} ln(\frac{E^t + p_z^t}{E^t - p_z^t}) \tag{6}$$

The inputs and their normalization are chosen by "norm". The inputs are:

- "s_pro": $\frac{1}{E_{cm}^{pro}}(p_x^t, p_y^t, p_z^t, E^t, p_z^{\bar{t}}, E^{\bar{t}})$
- "s_int": $\frac{1}{E_{cm}^{int}}(p_x^t, p_y^t, p_z^t, E^t, p_z^{\bar{t}}, E^{\bar{t}})$
- "cmass": $\frac{E_{cm}^{int}}{E_{cm}^{pro}}, y, \frac{1}{E_{cm}^{int}}(p_x^t, p_y^t, p_z^t)$

The nn architecture is defined by: 5 or 6 inputs; 2 dense layers with 16 nodes and activation function "relu"; the output layer with one node and a linear activation function.

The loss function is selected by "lossfunc" and can be the mean squared error or the chi square function, defined as $\chi^2 = \sum_i \frac{(s_i - w_i)^2}{w_i}$.

The nn is trained to predict respectively w_i or $-ln(w_i)$, depending on the selected "output". The training can early stop in the case of overfitting.

3 Definitions of maxima functions

The maxima functions are:

- "max_quantile_reduction", that is defined as in the paper. If one array is passed to this function, it finds the maximum s_max for the s_i, such that the events with larger weights than s_max (which become overweights) contribute to the total weights less or equal than the fraction r of the total sum. The formula is

$$s\_max = min\left\{ s_{\hat{i}} | \sum_{i=\hat{i}+1}^{N} s_i \leq r \sum_{i=0}^{N} s_i \right\} \tag{7}$$

  where s_i are ordered in an increasing order. If the parameter $r \leq 0$, then the maximum is $(1-r)$ times the real maximum. This is done to study the efficiencies in the case of no overweight. If two arrays are passed to this function (s_i and x_i arrays), then it find the maximum x_max respect the total $\sum_{i=0}^{N} s_i x_i$. While it behaves in the same way for $r \leq 0$

$$x\_max = min\left\{ x_{\hat{i}} | \sum_{i=\hat{i}+1}^{N} s_i x_i \leq r \sum_{i=0}^{N} s_i x_i \right\} \tag{8}$$

  where s_i and x_i are ordered respect the increasing order of x_i.

- "max_median_reduction"

Each maxima functions has a different array of r due to their different definitions.

4 Unweighting

There is a first loop on the index "i_r1" to test different values of the parameter r in the evaluation of s_max (or w_max). The nn computes the predicted weights of the validation data and save them in the array "s1". In the case of output=lnw, it takes the exponential of the negative predicted output as s1.

In the first unweighting it computes:

- s_max or x_max, respectively for new unweighting and paper unweighting. The parameters of this maximum are maxfunc (which defines the maxima function used) and the constant r.
- s2, which are the predicted weights of kept events. The new unweighting method unweights the events respect s_max such that

$$s2 = \{s1_i \mid \frac{s1_i}{s\_max} > rand1_i\} \tag{9}$$

  where $s1_i$ is the i-th event of the s1 array and $rand1_i$ is a uniform random number between 0 and 1 of the rand1 array. The paper unweighting method unweights the events respect w_max, like it is done in the paper, such that

$$s2 = \{s1_i \mid \frac{s1_i}{w\_max} > rand1_i\} \quad . \tag{10}$$

- z2, which are the "unweighted" weights after the first unweighting. In the new method

$$z2_i = sign(s2_i)max\left(1, \frac{|s2_i|}{s_{max}}\right) \tag{11}$$

  while, in the paper method

$$z2_i = sign(s2_i)max\left(1, \frac{|s2_i|}{w_{max}}\right) \tag{12}$$

- w2, which are the true weights of kept events.
- x2, which are the ratio between the true weights and the predicted one of the kept events

$$x2_i = \frac{w2_i}{|s2_i|} \quad . \tag{13}$$

- eff1, which is the efficiency of the first unweighting. In the new method it is $eff1 = efficiency(z2, s1, s\_max)$ and in the paper method it is $eff1 = efficiency(z2, s1, w\_max)$.

In the second unweighting there is a second loop on the index "i_r2" to test all the combinations of r between the one used for s_max (or w_max) and the one for x_max. Thus $s\_max = s\_max(i\_r1)$ and $x\_max = x\_max(i\_r1, i\_r2$, which means there is a different array of x_max for each value of s_max. The same is true also for efficiency 1 and efficiency2.

In the second unweighting it computes:

- x_max, which is the maximum of x2. In the new method is defined such that $x\_max = my\_max(x2_i \times |z2_i|)$, while in the paper method it is defined such that $x\_max = my\_max(x2_i)$.
- s3, which are the predicted weights of kept events. The new unweighting method unweights the events respect x_max such that

$$s3 = \{s2_i \mid \frac{x2_i}{x\_max} max(1, \frac{|s2_i|}{s\_max}) > rand1_i\} \quad . \tag{14}$$

  In the paper unweighting method

$$s3 = \{s2_i \mid \frac{x2_i}{x\_max} > rand2_i\} \quad . \tag{15}$$

- z3, which are the "unweighted" weights after the second unweighting. In the new method

$$z3_i = sign(z2_i)max\left(1, \frac{x2_i}{x\_max}max\left(1, \frac{|s2_i|}{s_{max}}\right)\right) = sign(z2_i)max\left(1, \frac{x2_i}{x\_max}|z2_i|\right) \quad . \tag{16}$$

  In the paper method

$$z3_i = max\left(1, \frac{|x2_i|}{x_{max}}\right) \quad . \tag{17}$$

- w3, which are the true weights of kept events.
- x3, which are the ratio between the true weights and the predicted one of the kept events
- eff2, which is the efficiency of the second unweighting. In both methods it is defined as $eff2 = efficiency(z3, x2, x\_max)$.
- f_eff, which is the effective gain factor. In the new method it is defined as $f\_eff = effective\_gain(z3, eff1, eff2)$, where $z3$ are the final "unweighted" weights. In the paper method it is defined as $f\_eff = effective\_gain(z2 \times z3, eff1, eff2)$, where $z2$ and $z3$ are the "unweighted" weights of first and second unweightings.

5 Plots of results

There are 3 plots for each test: training and evaluation (train.pdf); efficiencies and effective gain factor (eff.pdf); weights distributions (ws.pdf). In the eff pdf, the first plot represents eff_1 respect to s_max. The large dot represents the case where we s_max is the real maximum, so at his left there are $r > 0$ and at his right $r < 0$. In the second plot of eff.pdf, there are the eff_2 respect x_max. In this case the limit in x axis are set to (0, 12), to exclude large value of x_max that are present for the first indices of i_r1 (large overweight for s_max) with last indices of i_r2 (when x_max is 2 or 10 times the real maximum). In the third plot of eff.pdf there is the effective gain factor respect to s_max and x_max. In the label of y axis the x_max have an asterisk because they are not described by an array, but by a matrix. For simplicity I represented only x_max[i, i] in the label of the y axis, while I used their true value (that is x_max[i_r1, i_r2]) for all the computations.

In the first plot of ws.pdf there is the distribution of $x1 = w1/|s1|$ which are the true weights over the predicted weights before the unweighting. In this plot the x axis limits are not fixed: The second plot is equals to the first but with x axis limit fixed at (-0.5, 2.5). The third plot represents the distribution of x1 respect w1 with an histogram 2d, in this case the x axis limits (of w1) are not fixed, while the y axis limits (of x1) are fixed at (0, 2).