documentation nn_unwgt

lucabeccatini

October 7, 2022

1 Settings

The settings representing some features of the programs are:

- "seed_all": seed for tf and np.
- "norm": the normalization of the inputs of the nn. "s_pro" is the normalization respect the E_{cm} of proton, "s_int" is the normalization respect the E_{cm} of the interaction.
- "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.

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

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

  thus the Kish factor is: $\alpha = \frac{(\sum_i^N w_i)^2}{N \sum_i^N w_i^2}$. α is always lower than 1 if there are overweighted events and and equal to 1 if they are not.

- "efficiency_1": efficiency of the first unweighting defined as

$$eff_1 = \frac{\alpha_1 N_2}{N_1} = \frac{(\sum_i^{N_2} max(1, s_i/w_i)^2}{N_2 \sum_i^{N_2} max(1, s_i/w_i)^2} \frac{\sum_j^{N_1} s_i}{N_1 s_{max}} \tag{2}$$

  where N_1 is the initial number of events, N_2 is the number of events after the first unweighting, α_1 is the Kish factor for the first unweighting, s_i are the predicted weights, s_{max} is the maximum of s_i defined by the maxima function, w_i are the true weights.

- "efficiency_2": efficiency of the second unweighting defined as

$$eff_2 = \frac{\alpha_2 N_3}{N_2} = \frac{(\sum_i^{N_3} max(1, \frac{x_i}{x_{max}} max(1, \frac{s_i}{s_{max}}))^2}{N_3 \sum_i^{N_3} max(1, \frac{x_i}{x_{max}} max(1, \frac{s_i}{s_{max}}))} \frac{\sum_j^{N_2} x_i}{N_2 x_{max}} \tag{3}$$

  where N_3 is the number of events after the second unweighting, α_2 is the Kish factor of the second unweighting, $x_i = w_i/s_i$ are the ratio between the true weights and the predicted value, x_{max} is the maximum of x_i defined by the maxima function.

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

$$f\_eff = \frac{N \frac{t_{st}}{\epsilon_{st}}}{\frac{N}{\alpha}(\frac{t_{nn}}{\epsilon_1 \epsilon_2} + \frac{t_{st}}{\epsilon_2})} = \frac{\alpha}{\frac{t_{nn}}{t_{st}} \frac{\epsilon_{st}}{\epsilon_1 \epsilon_2} + \frac{\epsilon_{st}}{\epsilon_2}} \tag{4}$$

  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, $\epsilon_{st} = eff1\_steff2\_st$ 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.

2 Prediction of weights

Firstly, it reads the momenta and weights of events from "info_wgt_events_5iter.txt", where in each line there are: $p_x^t, p_y^t, p_z^t, E^t, p_z^{\bar{t}}, t^{\bar{t}}, weight$.

The the momenta are normalized respect the chosen normalization ("norm"). "s_pro" divide each momentum by the same quantity, while "s_int" divide the momenta of each event by the E_{cm} of that event.

The nn architecture is defined by: 6 inputs, which are the momenta; 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

$$s\_max = min\left\{ s_{\hat{i}} | \sum_{i=\hat{i}+1}^{N} s_i \leq r \sum_{i=0}^{N} s_i \right\} \tag{5}$$

  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 $r \leq 0$

$$x\_max = min\left\{ x_{\hat{i}} | \sum_{i=\hat{i}+1}^{N} s_i x_i \leq r \sum_{i=0}^{N} s_i x_i \right\} \tag{6}$$

  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 all the different values of the parameter r in the evaluation of s_max. The nn computes the weights of the validation data and save them in the array "s1". In the case of output=lnw, we take the exponential of the negative predicted output as s1. Then it finds s_max through "my_max" and it performs the first unweighting, saving the kept weights in the "s2" array and their true weight in "w2". It compute the ratio "x2" between the true weight of kept events and the absolute value of their predicted value

$$x_i = \frac{w_i}{|s_i|} \quad . \tag{7}$$

The first unweighting ends with the computation of efficiency 1.

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 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. It computes x_max and performs the second unweighting saving the kept events on s3 and x3. Finally it computes the efficiency 2 and the effective gain factor.

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).