Monte Carlo tuning with Neural Networks

S. Alioli, S. Carrazza

December 6, 2016

1 Idea

Determine the best set of input parameters, $\vec{p}_{\rm in}$, and its uncertainties for Monte Carlo tunes through artificial neural networks.

2 Methodology

Here we discuss two methods which should be interesting to study.

2.1 Strategies

2.1.1 χ^2 model

- Take the input MC histograms $H_{\rm MC}(\vec{p}_{\rm in})$ and build MC bootstrap replicas based on the MC statistical uncertainties per bin, $H^k_{\rm MC}(\vec{p}_{\rm in})$ with $k = [1, N_{\rm rep}]$.
- For each MC replica we compute

$$\chi^2_k(\vec{p}_{\rm in}) = \left[\sum_{ij}(d_i - H^k_{\rm MC}(\vec{p}_{\rm in})_i)({\rm cov}^{-1})_{ij}(d_j - H^k_{\rm MC}(\vec{p}_{\rm in})_j)\right]^{\frac{1}{2}}, \quad (1)$$

  where d_i is the experimental data and cov its covariance matrix.
- At this stage we have $N_{\rm rep}$ replicas for the $\chi^2_k(\vec{p}_{\rm in})$ function so we apply a gradient or genetic algorithm minimization to get the best $\vec{p}^{\,k}_{\rm in}$ for each replica which satisfies the condition:

$$\chi^2(\vec{p}_{\rm in}) \to 0 \quad (2)$$

2.1.2 Monte Carlo model

In this method we take the input MC histograms $H^i_{\rm MC}(\vec{p}_{\rm in})$ and build a model which maps the input variables $\vec{p}_{\rm in}$:

- for each MC histogram we build a neural network model $NN(\vec{p}_{\rm in}, i) \rightarrow H^{i}_{\rm MC}(\vec{p}_{\rm in})$ by generating bootstrap MC replicas first and then fitting the neural network to the target MC histogram including its uncertainties in the loss function.

- build a χ^2 function which measures the distance between the target data and the MC NN model.

2.2 Validation

Perform a closure test in order to check the quality and efficiency of the procedure.