Efficiency modelling for ultra-fast simulation

Lucio Anderlini

Istituto Nazionale di Fisica Nucleare
SEZIONE DI FIRENZE

Introduction

In the ultra-fast simulation of the muon identification at LHCb we need to simulate:

- The likelihood of the detector response (see Gabriella's presentation)
- The hardware-computed, trigger-accessible "*IsMuon*" condition

IsMuon is a boolean flag associated to each track, obtained counting the number of *hits* in the muon detector in *Fields of Interests:* large OR-combinations of pads of the muon detector.

The IsMuon condition can be obtained from FPGAs directly at the hardware level of the trigger (L0-trigger) and introduces an efficiency on the muon identification that we wish to model and **include in the parametric simulation.**

Some formalism

Given the momentum *p*, the pseudorapidity η, and the detector occupancy encoded in the number of tracks (`nTracks`), we wish to parametrize the efficiency

$$\varepsilon_{\texttt{isMuon}} = \varepsilon(p, \eta, \mathrm{nTracks})$$

which is **the average of the Bernoulli distribution describing the probability that a particle with given *p*, η and `nTracks` has `isMuon` == 1.**

Given a sample of particles

$$X = \{x_i\}_i = \{(p_i, \eta_i, \texttt{nTracks}_i)\}_i$$

we can compute the `isMuon` probability distribution of the single particle as

$$P(x_i|\varepsilon_{\texttt{isMuon}}) = \begin{cases} \varepsilon_{\texttt{isMuon}}(x_i) & \text{for } \texttt{isMuon}(x_i) = 1 \\ 1 - \varepsilon_{\texttt{isMuon}}(x_i) & \text{for } \texttt{isMuon}(x_i) = 0 \end{cases}$$

Likelihood and cross-entropy

For definition of *Likelihood:*

$$\mathcal{L}(\varepsilon_{\text{isMuon}}|X) = \prod_i \mathrm{P}(x_i|\varepsilon_i)$$

Replacing the probability function:

$$\mathcal{L}(\varepsilon_{\text{isMuon}}|X) = \prod_{\texttt{isMuon==1}} \varepsilon_{\texttt{isMuon}} \cdot \prod_{\texttt{isMuon==0}} (1-\varepsilon_{\texttt{isMuon}})$$

$$\log\mathcal{L}(\varepsilon_{\text{isMuon}}|X) = \sum_{\texttt{isMuon==1}} \log(\varepsilon_{\texttt{isMuon}}) + \sum_{\texttt{isMuon==0}} \log(1-\varepsilon_{\texttt{isMuon}})$$

$$\boxed{\log\mathcal{L}(\varepsilon_{\text{isMuon}}|X) = \sum_i \left[\texttt{isMuon}\cdot\log(\varepsilon_{\texttt{isMuon}}) + (1-\texttt{isMuon})\cdot\log(1-\varepsilon_{\texttt{isMuon}})\right]}$$

Bernoulli cross-entropy × (-1)

Parametrizing the efficiency

We define a parametrization of the efficiency through the parameters θ

$$\varepsilon_{\texttt{isMuon}} = \varepsilon_{\texttt{isMuon}}(p, \eta, \texttt{nTracks}; \theta)$$

So that the likelihood becomes:

$$\log \mathcal{L}(\varepsilon_{\text{isMuon}}(p, \eta, \texttt{nTracks}; \theta) | X) = \sum_i \left[\texttt{isMuon} \cdot \log(\varepsilon_{\texttt{isMuon}}(p, \eta, \texttt{nTracks}; \theta)) + (1 - \texttt{isMuon}) \cdot \log(1 - \varepsilon_{\texttt{isMuon}}(p, \eta, \texttt{nTracks}; \theta)) \right]$$

We can then obtain the optimal choice of the parameters θ as

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \, \mathcal{L}(\theta | X)$$

This result is totally independent on the choice we make on the parametrization.

Since we want something quick and easy to minimize, we parametrize it with a neural network.

The neural network

The neural network used in this example is a single-hidden-layer perceptron with a 10-neuron hidden layer and *tanh* activation function.

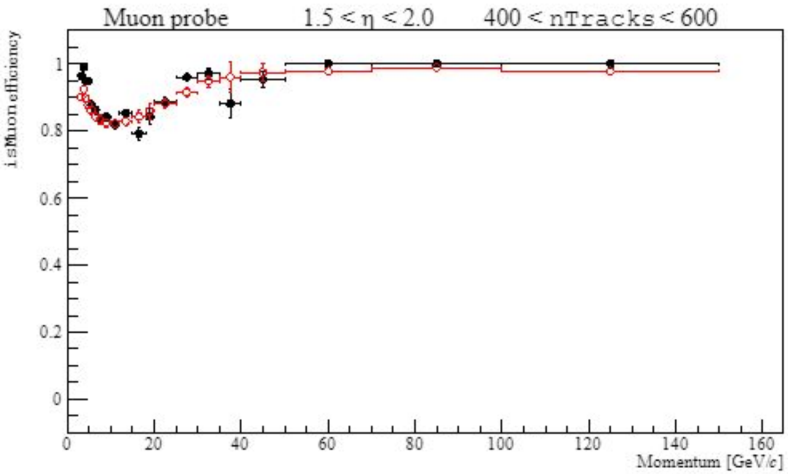
The output layer is activated through a *sigmoid* function to ensure the output is constrained to the interval (0,1).

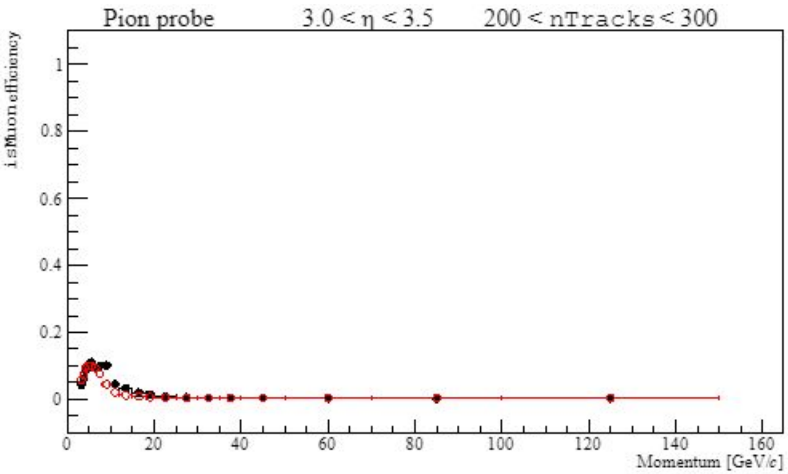
To avoid numerical divergence of the logarithms in the loss function, we impose that if $\varepsilon_{\mathtt{isMuon}} < 10^{-12}$, then 10^{-12} is taken.

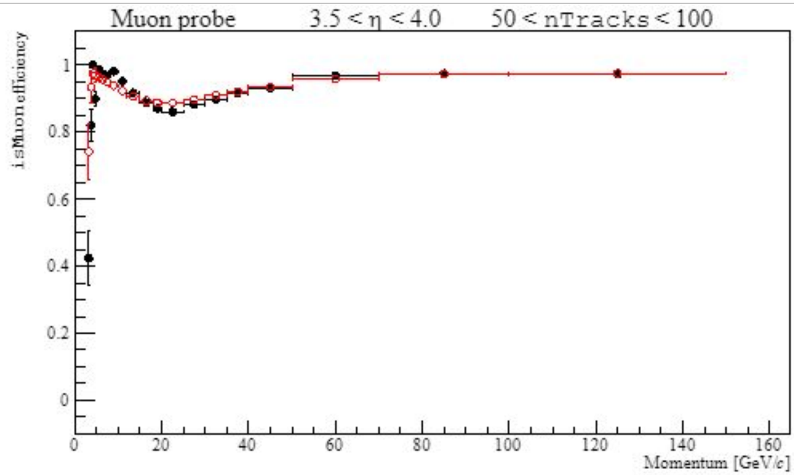
The training of the neural network requires less than one hour (mainly because there are a lot of data).

Some results

Some comparison between the efficiency of the isMuon requirement in real data (black markers) and as obtained from the trained neural network (red empty markers).







How to apply the neural network to simulated data

When possible, the best option is to **weight each event by its efficiency**.
This allows to preserve the total statistical precision of the original sample while describing the shapes of *signal* and *misidentification background* contributions.

When integrated within the official LHCb simulation which does not allow to define weights for the reconstructed tracks, we draw a random number r from a uniform distribution and, if $r < \varepsilon_{\mathtt{isMuon}}$ we consider the particle as identified as muon.

Conclusion

We have described a technique to parametrize the selection efficiency of the binary criterion *isMuon* with a simple neural network.

The method is more general and can be applied to several other cases where an efficiency must be modeled as a function of some variables.

The parametrized efficiency for `isMuon` is integrated within the simulation framework of LHCb, but can also be used to define shapes and misidentification rates by weighting the events for their efficiency.