Report

A study of machine learning algorithms for reconstruction of missing energy in particle physics experiments.

Max Isacson, max.isacson@physics.uu.se
Mikael Mårtensson, mikael.martensson@physics.uu.se
Camila Rangel Smith, camila.rangel@physics.uu.se
Henrik Öhman, ohman@cern.ch

March 31, 2016

1 Introduction

2 The Dataset

2.1 Structure

The dataset consists of simulated pp collision events, in which a charged Higgs is produced and decays as $H^+ \to \tau\nu$. Each event is described by one set of observable variables and one set of unobservable variables.

Observable variables:

- E_x^{miss}, E_y^{miss}— The x- and y-components of the missing energy.
- $P_{\tau_{\text{vis.}}}$ — The 4-momentum of the visible (hadronic) part of the τ decay.
- P_{b_0}, P_{b_1}, P_{q_0}, P_{q_1} — The 4-momenta of the two b-jets and two light jets.

Unobservable variables:

- P_{ν_τ} — The 4-momentum of the neutrino from the charged Higgs decay.
- $P_{\bar{\nu}_\tau}$ — The 4-momentum of the neutrino from the τ decay.

2.2 Production

MG5_aMC@NLO [1] is used for the matrix element computation of $gg/q\bar{q} \to H^+$ and the event simulation. The events are then passed to PYTHIA8 [2] for the showering and hadronization, and for the $H^+ \to \tau\nu$ decay. Finally, the detector response is simulated using DELPHES [3] with an ATLAS-like geometry.

2.3 Estimated and true quantities

The true quantities for both the observable and the unobservable variables are available as output from the event simulation. The estimated values for the observable variables are reconstructed from the output from the detector response simulation.

3 Solution method

3.1 Predictor selection

3.2 Neural Network

3.2.1 Data scaling

In this study, the dataset was scaled such that the training set mean was set to 0 and the standard deviation 1 to for each input variable x_i and the training target t_{tr}:

$$x'_i = \frac{x_i - \mu(x_{i,\mathrm{tr}})}{\sigma(x_{i,\mathrm{tr}})}, \tag{1}$$

$$t'_{\mathrm{tr}} = \frac{t_{\mathrm{tr}} - \mu(t_{\mathrm{tr}})}{\sigma(t_{\mathrm{tr}})}. \tag{2}$$

This improves the training performance by avoiding the flat tails of the activation functions. The reverse of equation 2 is then applied to the Neural Network (NN) prediction of the validation t'_{val} to get the physical output t_{val}:

$$t_{\mathrm{val}} = t'_{\mathrm{val}}\sigma(t_{\mathrm{tr}}) + \mu(t_{\mathrm{tr}}) \tag{3}$$

3.2.2 Implementation

The python package Keras [4] was used to implement neural networks. The number of input nodes is restricted to the number of predictor input variables (25) and the output to the target dimension of 1. Also, since the network is used for regression, the output activation function is set to a linear function. A large number of configurations of the hidden layers were tested by varying the number of layers, the number of perceptrons in each layer, and the activation functions (sigmoid, tanh, softmax, and relu).

4 Results

References

[1] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H. S. Shao, T. Stelzer, P. Torrielli, and M. Zaro. The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. *JHEP*, 07:079, 2014.

[2] Torbjörn Sjöstrand, Stefan Ask, Jesper R. Christiansen, Richard Corke, Nishita Desai, Philip Ilten, Stephen Mrenna, Stefan Prestel, Christine O. Rasmussen, and Peter Z. Skands. An introduction to PYTHIA 8.2. *Computer Physics Communications*, 191:159 – 177, 2015.

[3] J. Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, and M. Selvaggi. Delphes 3: a modular framework for fast simulation of a generic collider experiment. *Journal of High Energy Physics*, 2014(2):1–26, 2014.

[4] Keras: Deep learning library for theano and tensorflow. `http://keras.io/`. Version 0.3.2-1.