Activity Report

July 2018 - June 2019

Silvio L. Stanzani

Projeto 2597/2017 – Centro de Excelência em 'Machine Learning'

Center for Scientific Computing
São Paulo State University (UNESP)

September 2019

1. Introduction

The Center of Excellence on Machine Learning has the objective of creating a research center to work on challenging projects related to Machine Learning, and also to attack problems with scientific relevance and/or social and economic impact. My activities on this initiative have been centered on the investigation of how to use Machine Learning techniques in order to tackle practical problems on the field of High Energy Physics (HEP), more specifically problems related to reconstruction of particle trajectories on detectors.

This activity report seeks to outline and present the activities I have performed from July 2018 to June 2019.

2. Activities Performed

During the period of this project i performed the following activities:

- The preparation of an introductory course to machine learning in order to identify the opportunities to apply machine learning to HEP challenges (presented in section 2.1).
- I investigated the viability to use machine learning for performance prediction on heterogeneous architectures (presented in section 2.2)
- finally, I developed a model based on Neural Network to classify tracks from particle detector as real or fake (presented in section 2.3).

2.1 Basic Machine Learning Course

I have developed an introductory machine learning tutorial covering the following topics:

- Introduction to Statistical Learning
- Regression Analysis
- Linear Regression
- Logistic Regression
- Introduction to Neural Networks

• Introduction to Convolutional Neural Networks

2.2 Performance Prediction using Machine Learning

Predicting the performance of parallel applications in different computational architectures is essential to improve clusters' utilization, providing insights about the design of better job schedulers. In order for these schedulers to be able to make runtime decisions, two main constraints must be considered:

1) The application is provided without the source code and
2) The prediction must be made within a short time.

In order to implement such prediction i decided to make a statistical model based on information provided by profiler, because it is capable of measuring several aspects of application's performance, and also on the Roofline model that is capable of model the performance with a high level of abstraction.

The statistical model was developed using a set of applications with different characteristics, so we evaluated the model with the objective of capture this characteristics and thus predict the performance on heterogeneous architectures.
As a result of this activity I published the following paper with other colleagues from NCC on Vecpar 2018.

"Towards as Strategy for Performance Prediction on Heterogeneous Architectures"
Authors : Silvio Stanzani, Raphael Cóbe, Jefferson Fialho, Rogério Iope, Marco Gomes, Artur Baruchi and Júlio amaral
published as LNCS (Lecture Notes in Computer Science) and presented in 13th International Meeting on High Performance Computing for Computational Science (Vecpar 2018).

Also I have presented this work as an invited talk at the Fourteenth International Workshop on Automatic Performance Tuning[1].

[1] http://www.iwapt.org/2019/

2.3 ML for HEP

The objective of this project is to tackle the challenge of reconstruct particle tracks from hits left in the silicon detectors, this objective was inspired in a kaggle challenge[2].

Several hits are collected from detectors that has to be combined to a single tracks and connected to the original particle. Nowadays, this process is carried out using Kalman filter, in this project we will build a machine learning model based on neural networks to reconstruct the tracks.

In the next sessions, we are going to describe the procedures for data preparation and the neural network topology for classify tracks as real or fake.

2.3.1 Data Preparation

The files from kaggle database are available at **/data/trackMLDB/train**, and can be manipulated using the track-ml library[3]. The dataset is composed by 4 tables:

- hits
- cells
- particles
- truth

One track is a composition of one particle from **particle** table, several lines from **hits** table and each hit can have one or more events from **cells** table. The following figure shows the data organization.

In order to convert to organize data in a way that could be possible to be used as input to neural networks we used the following script[4] to enroll tracks into single lines. The table is composed by six columns for particle and 28 sets of six columns for each hit. If track has less than 28 hits the subsequent hits are filled with zeros. Each hit can

[2] ttps://www.kaggle.com/c/trackml-particle-identification
[3] https://github.com/LAL/trackml-library
[4] https://github.com/silviostanzani/track-ml-1/blob/master/utils/tracks_manipulation_lib.py

have one or more event of channel, so to represent that information in a single line we obtained the average of all ch0, ch1 and weight values. The following figure represents the input table.

Particle						Hit					
tx	ty	tz	px	py	pz	tx	ty	tz	ch0	ch1	w

The kaggle challenge provides five set of tracks, we enrolled all the tracks in the following files:

- /data/trackMLDB/analysis/train_1_realv3
- /data/trackMLDB/analysis/train_2_realv3
- /data/trackMLDB/analysis/train_3_realv3
- /data/trackMLDB/analysis/train_4_realv3
- /data/trackMLDB/analysis/train_5_realv3

We also have auxiliary files that represent subsets of the aforementioned:

- /data/trackMLDB/analysis/pt1p0_train_[N]_realv3, N = 1 to 5, is a subset of the corresponding file whilst keeping only particles with transverse momentum pt > 1.0 GeV
- /data/trackMLDB/analysis/pt2p0_train_[N]_realv3, N = 1 to 5, is a subset of the corresponding file whilst keeping only particles with transverse momentum pt > 2.0 GeV

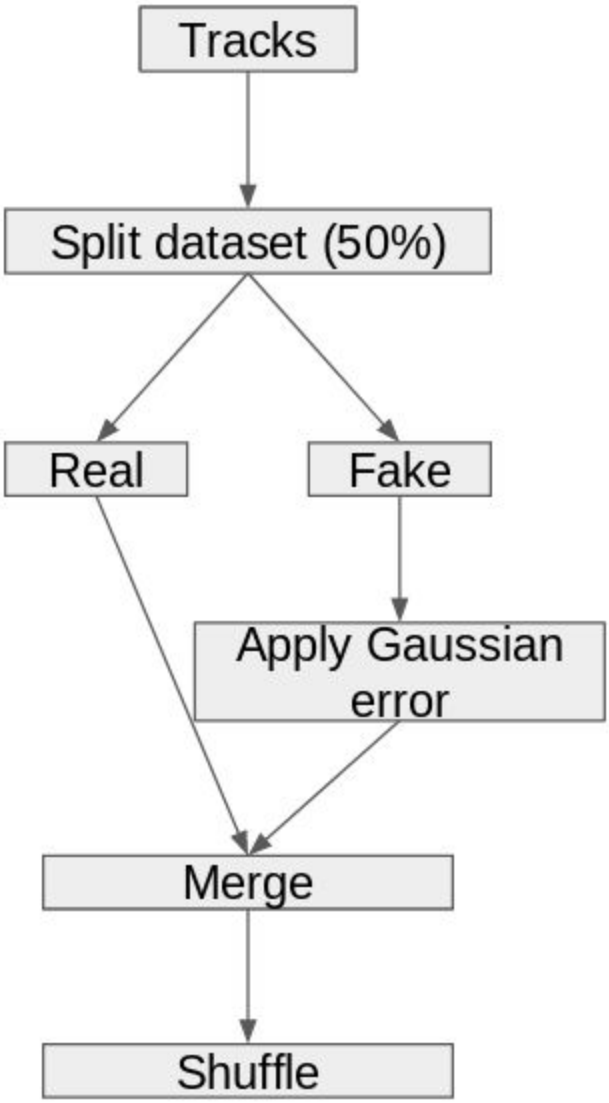
In order to prepare the input data to the neural network that classify track as real or fake we developed a script that generates fake tracks from real tracks[5]. This script receives one real track, generates a gaussian noise in random hits, and returns one fake track.

Based on this script we create another script that takes a set of real tracks and generates a new file with the same number of real and fake tracks, performing the following steps:

- Cuts the first 6 columns that is information about particle and for each hit cuts the information about channel 0, channel 1 and weight;
- The 3D position information is modified using the function that applies a gaussian function to modify the X,Y and Z value. The parameter called **err** defines the range of gaussian error that will be applied;

[5] https://github.com/silviostanzani/track-ml-1/blob/master/utils/tracktop.py

- The fake tracks created are merged and shuffled.



2.3.2 MLP for classify tracks as real or fake

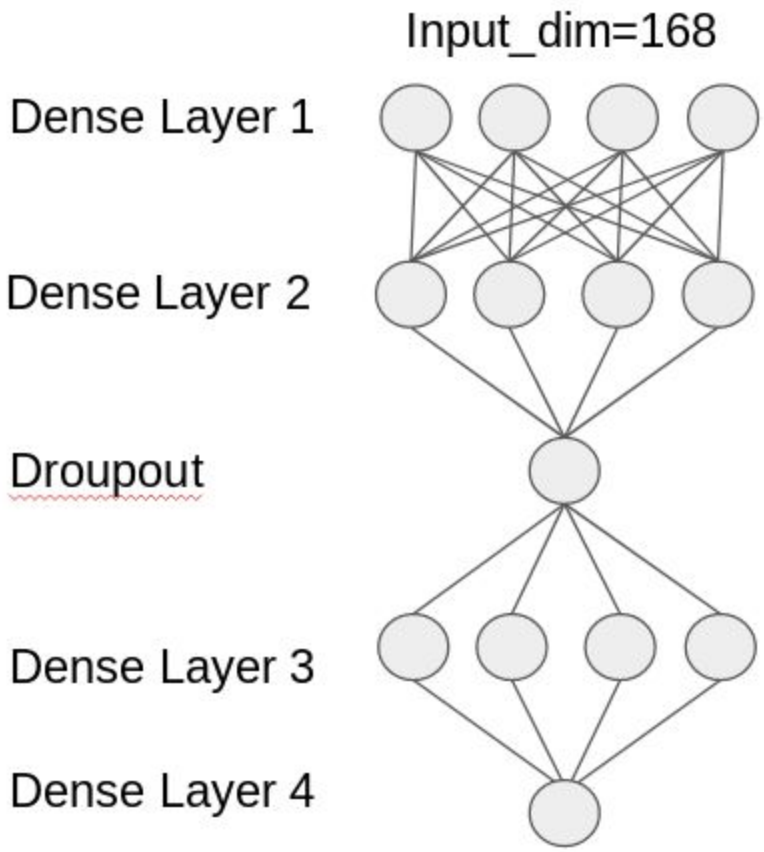
We developed a Multi Layer Perceptron (MLP) to perform binary classification of tracks as fake or real. In order to identify the best set of parameters we developed a script that evaluate several parameters and returns the set of parameters that presents the best accuracy [6].

[6] https://github.com/SPRACE/track-ml/blob/master/Hyperparameter_search_for_track_classifier.py

This script returned the following set of parameters:

- activation = 'hard_sigmoid'
- Neurons = 4
- Init_mode = 'he_normal'
- opt = 'Adadelta'

The MLP receives a track with any number of hits and returns 0 for real and 1 for fake. The following figure shows the network topology.



3. Conclusions

This report showed the results achieved during the period from july 2018 to june 2019 of activities related to the scholarship in the context of Projeto 2597/2017 – Centro de Excelência em 'Machine Learning'.