Proceedings of CTD 2020
PROC-CTD2020-36
September 29, 2020

Exploring (Quantum) Track Reconstruction Algorithms for non-HEP applications

K.Novotny[1], D. Dobos[1,2], K. Potamianos[1,3], C. Tüysüz[4,5], B. Demirköz[5], F. Carminati[6], S. Vallecorsa[6], F. Fracas[6,7], J.-R. Vlimant[8]

[1]*gluoNNet, Geneva, Switzerland*
[2]*Lancaster University, Lancaster, UK*
[3]*DESY, Hamburg, Germany*
[4]*Middle East Technical University, Ankara, Turkey*
[5]*STB Research, Ankara, Turkey*
[6]*CERN, Geneva, Switzerland*
[7]*University of Padua, Padua, Italy*
[8]*California Institute of Technology, Pasadena, California, USA*

ABSTRACT

An increase in simultaneous collisions is expected creating a challenge for accurate particle track reconstruction in High Luminosity LHC experiments. Similar challenges can be seen in non-HEP trajectory reconstruction use-cases, where tracking and track evaluation algorithms are used. High occupancy, track density, complexity and fast growth therefore exponentially increase the demand of algorithms in terms of time, memory and computing resources. While traditionally Kalman filter (or even simpler algorithms) are used, they are expected to scale worse than quadratically and thus strongly increasing the total processing time. Graph Neural Networks (GNN) are currently explored for HEP, but also non-HEP trajectory reconstruction applications. Quantum Computers with their feature of evaluating a very large number of states simultaneously are therefore good candidates for such complex searches in large parameter and graph spaces. In this paper we present our work on implementing a quantum-based graph tracking machine learning algorithm to evaluate Traffic collision avoidance system (TCAS) probabilities of commercial flights.

PRESENTED AT

Connecting the Dots Workshop (CTD 2020)
April 20-30, 2020

1 Introduction

While conventional machine learning techniques have the ability to deal with today's generated amount of data succinctly, it is interesting to explore if novel techniques such as quantum computing have the possibility to outperform existing techniques. This possibility and the analogy between track reconstruction in particle physics and aviation has led to a collaboration between researchers at gluoNNet, CERN openlab, Middle East Technical University Ankara, Caltech and DESY. There are striking similarities between particle track reconstruction and tracking of air plane routes. In both cases, there is a huge increase of the amount of data expected, in particle physics during the upcoming High Luminosity run of the LHC [1], in aviation due to the use of ADS-B [2] emitters and an expected explosion in the use of drones. Particular similarities between particle physics and aviation are for instance the detector hit and the broadcasted timestamp of the airplane as both include position and time information. In this scenario, particles correspond to airplanes, an event Id in physics to an airplane's transponder Id and the particle Id's to the aircraft type. Track reconstruction of airplanes has in this study two major aims: understanding incidents and understanding triggers of a deviation from their own flight path compared to the optimal route. As a next step, these tracking algorithms could be applied to online data for predicting flight paths based on the available data. These predictions could be used for optimisation of airport capacities for starting and landing and for improving flight safety, leading to a better prediction of emergency situations and an earlier warning than the Traffic Collision Avoidance System (TCAS) currently provides.

2 Dataset

The dataset used in this study contains the CAT 21 information [3] related to civilian aircraft usages, covers the month of April 2019 and has been kindly provided by Aireon [4] and the Irish Aviation Authority [5]. It covers a month's worth of data that correspond to 500 GB, which is relatively small compared to data sets used in high energy physics. At the moment, not every aircraft is equipped with an ADS-B device broadcasting information to the ground station. The dataset includes information such as the unique aircraft identifier and its position in degrees. The aircraft's position broadcasted in degrees of Longitude and Latitude is used in this study as well as the geometric height defined as the projected height to the earth's ellipsoid in WGS-84 coordinates [3]. There are three additional parameters relevant for the analysis, which are the type and weight-class of the aircraft referred to as emitter category as well as the aircraft's flight number. However, the latter is only available if it is known prior to departure and computed in the aircraft's onboard computer. The Quantum Graph Neural Network (QGNN) uses this information for the reconstruction of the aircraft's flight path.

3 Quantum Graph Network

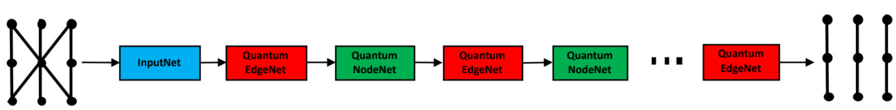
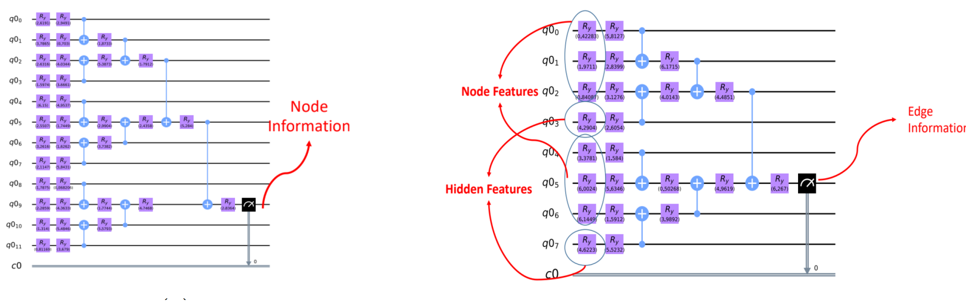


Figure 1: Structure of iterative application of the Quantum Node Network and Quantum Edge Network. Pictures taken from [8].

The underlying classical graph neural network (GNN) from [6] consists of an edge neural network and a node neural network and are applied iteratively after each other. In this work, both edge network and node network have been rewritten as quantum circuits via Tensor Tree Networks (TTN). These are chosen for simplicity and implemented using Pennylane [7] providing the necessary gradients of the Quantum circuits during the training. The Tensorflow interface is used for optimisation and constructing the model pipeline [9].



(a) (b)

Figure 2: Quantum Node Network (a) and Quantum Edge Network (b) are composed of Tensor Tree Networks (TTN). The underlying TTNs differ in the amount of qubits. 6 qubits are used to cover two sets of three spatial coordinates corresponding to two edges. The hidden features correspond to classical neural network layers. The obtained output of the Quantum Edge Network serves as input for the Quantum Node Network. Numerical values are example values. Pictures taken from [8].

The graph data consists of the timestamp, and the graph matrices indicate the relation between the different emitter Id's. A graph contains information about the relation of two spatial points, which are in total 6 coordinates. The graph data set is then ingested into an Input layer network - a single layer neural network - to increase the dimension of the input. This is necessary to obtain the number matching to the three spatial coordinates. Then, the graph data serve as the input of the hybrid Quantum Graph Neural Network (QGNN) consisting of Quantum Edge Network and Quantum Node Network that are applied iteratively and visualised in Figure 1.

3.1 Quantum Edge Network and Quantum Node Network

The Quantum Edge Network (QEN) consisting of 6 qubits is applied to each graph containing information of two edges described by two times three spatial coordinates. The output edge feature is passed into every edge to update their values. Their coordinates are then mapped in the physics case to $[0, 2\pi]$, whereas in the case of aircrafts their broadcasted position in Longitude and Latitude is used. This output is ingested into the Quantum Node Network (QNoN) and applied to each node of the graph for updating the hidden layers. The QNoN is applied to all possible neighbours that might be connected through the available edges. Both network architectures are visualised in Figure 2.

3.2 Tensor Tree Network

The TTN of the QEN differs from the TTN used for QNoN in the amount of qubits. There are 6 qubits in case of the QEN and each qubit corresponds to one spatial parameter. Additionally, there are two qubits representing hidden layers corresponding to classical neural network layers. $R_y(\theta)$, the rotation gate for performing around the Y axis of the Bloch shpere is then applied to encode the information of the spatial coordinates into the qubit.

$$R_y(\theta)|\theta\rangle = \cos(\theta/2)|0\rangle + \sin(\theta/2)|1\rangle \tag{1}$$

In case of the QEN, a measurement of one qubit is taken as displayed in Figure 2. For calculating the expectation value, the results of multiple measurements are then taken into account. This outcome is then the new input for the Edge feature. The same procedure is applied in case of the QNoN. For optimisation purposes of the variable in the TTNs in both QEN and QNoN, Tensorflow's ADAM optimiser [9] is used. The ADAM optimiser is a stochastic gradient descent method based on adaptive estimates for lower-order measurements [10].

4 Conclusions

This quantum computing simulation represents the initial stage of a track reconstruction algorithm that could be used in aviation for the purpose of improving the Traffic collision avoidance system (TCAS). Analogous simulations in particle physics show that this model performs well [8]. Further developments of the model and circuit are needed as this case only represents a static and a procedural case. The applicability of such a QGNN does not yet outperform available classical approaches due to the limited access to real quantum computers.

ACKNOWLEDGEMENTS

K.N., D.D., and K.P. thank Peter Griffiths and Jeff Ashbolt for their support and fruitful discussions as well as Aireon and the Irish Aviation Authority for providing the data. K.N. is grateful to A. Salzburger for fruitful discussions during the early stages of this project.

References

[1] G. Apollinari, et al. "High Luminosity Large Hadron Collider HL-LHC," CERN Yellow Report **2015-005**, 1-19 (2017) [arXiv:1705.08830 [physics.acc-ph]].

[2] ADS-B Exchange `https://www.adsbexchange.com/data//`.

[3] EUROCONTROL Specification for Surveillance Data Exchange ASTERIX Part 12 Category 21 ADS-B Target Reports, EUROCONTROL-SPEC-0149-12, `https://www.eurocontrol.int/publication/cat021-eurocontrol-specification-surveillance-data-exchange-asterix-part-12-category-21`

[4] Aireon-SPACE-BASED ADS-B `https://aireon.com/`.

[5] Irish Aviation Authority. `https://www.iaa.ie/`.

[6] S. Farrell, et al. "Novel Deep Learning Methods for Track Reconstruction," [arXiv:1810.06111 [hep-ex]].

[7] E. Grant, et al. "PennyLane: Automatic Differentiation of Hybrid Quantum-Classical Computations," [arXiv:1811.04968 [quant-ph]].

[8] C. Tüysüz, et al. "Particle Track Reconstruction with Quantum Algorithms," [arXiv:2003.08126 [quant-ph]].

[9] M. Abadi, et al. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," `https://www.tensorflow.org/`, (2015).

[10] Diederik P. Kingma and Jimmy Ba "Adam: A Method for Stochastic Optimization," [arXiv:1412.6980 [cs]]