Particle Jet Multi-Class Classification via Deep Neural Network Architecture

Nathan Roberts, Sharmi Mathur, Darren Chang

Data Science, University of California San Diego, La Jolla, USA
March 7, 2021

Abstract

As data scientists, we are often driven toward those domains which generate vast amounts of data. High-energy physics is no exception. The Large Hadron Collider (LHC) alone produces around 90 petabytes of data per year (roughly 240 terabytes per day). As such, there are thousands upon thousands of researchers combing through the LHC's particle interactions to draw conclusions. But, there exists one major difficulty in doing so: the colliders themselves only have instruments that can detect physical quantities (energies, momentums, and the like). The LHC simulates particle collisions that result in a spray of subatomic particles called jets. Considering the many categories of jets (Higgs boson, singly charmed quarks, etc.), classification of jets must be conducted outside of the LHC by researchers and their algorithms.

We implement multiple multiclass classifiers (CNN, GNN, ENN) to discriminate between six types of jets which may occur. While a similar classifier exists at the LHC, the ceiling for improvement extends higher with each advancement in machine learning- deep neural network architecture being the most recent. In implementing our own neural network, we strive to achieve a higher level of model performance.

1 Introduction

In order for there to be study of subatomic particles, and indeed for any knowledge to be gained about the quantum world at all, physicists must use particle colliders. The Large Hadron Collider (LHC) at CERN produces data on the order of fifty petabytes a year (expected to increase further with newer updates to the collider), making high energy physics (HEP) data very appealing for data scientists like ourselves. These devices accelerate opposing beams of protons along a track until they collide with velocities just shy of the speed of light. The impact then forces each proton in the collision to scatter into the subatomic, elementary particles which compose it. This resulting spray of quarks, leptons, and bosons decay in a cone-shaped pattern referred to as a jet.

One of the limitations of particle colliders lies in the fact that there does not currently exist a magical device capable of simply detecting the presence of specific varieties of particles. Any determination of particle type and trajectory must be extrapolated from the detector's physical measurements. As each research team working with the data has a separate goal, whether it be to investigate the potential conditions of the early universe or the nature of the Higgs field and nature of matter as a whole, they will want to classify different types of particle jets. Drawing on past work [1] creating a neural network classifier to detect the presence of Higgs boson jets, we set out to create a multiclass classifier for six categories of jets representing different decay patterns of elementary particles.

The data collected represents fully simulated LHC collision events, released by the CMS Collaboration on the CERN Open Data portal [2]. These provided simulations allow for a more intrinsic, realistic comparison of machine learning methods on high-energy physics experiments. Considering our goal is to distinguish six different categories of particle jets from proton-proton collisions ($H\rightarrow bb$, $QCD\rightarrow b$, $QCD\rightarrow bb$, $QCD\rightarrow c$, $QCD\rightarrow cc$, and QCD_other), we particularly focus on the features representing jets (Fig 1).

Our dataset then provides metrics concerning the jet overall, individual tracks of particles within that jet, and any secondary vertices (locations from which particle tracks originate which are not the point of collision) which may be present. In total, there are 172 features for us to pick from when doing our analysis, of which 74 are specifically about the jet as a whole. Some key features that we are concerned with include number of tracks in a jet, the angles which define the physical morphology of the jet, number of secondary vertices present, and the

distance between the primary vertex and any secondary vertices present.

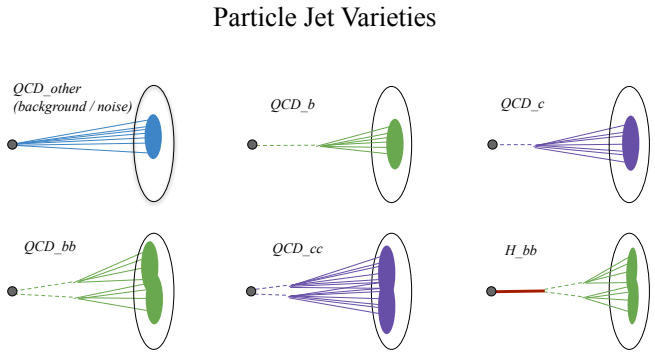

Figure 1. Visual representation of particle jets

2 Method

In our initial exploration, we used a convolutional neural network developed by the CMS Collaboration as a basis for creating a H→bb jet classifier of our own. Our replication followed similar model architecture to the original but with some slight variations. The final model uses 48 track features for up to 60 charged particles to draw conclusions. As this classifier was fairly competent at discriminating between those jets which contained a H→bb decay and those that did not, we adapted this to be our first multi class classifier.

This model utilizes the Conv1D layer of Keras, adding multiple 1D convolutional layers. Essentially, we are applying the Deep Sets [3] architecture to jets, known as the particle-flow network [4] approach. After batch normalization [5] on the input data, the features are passed to 3 separate one-dimensional convolution layers which build upon each other sequentially. The number of nodes in each layer are 64, 32, and 32, respectively. The outputs of these nodes are average pooled and then sent to a hidden dense layer with 100 nodes. This is finally passed to a final fully connected layer with 2 nodes which classifies the jet with a softmax activation function. All layers before this had ReLU [6] activations. As a baseline, we compare this against a naive, fully-connected, dense neural network. We refrain from using this architecture to make a final model with, as these kinds of fully connected neural networks are prone to overfitting.

Performing multi class classification is hindered often by the imbalance in class representation in the data. This is very much the case with our work, as should be evident from the distribution of class representation from our exploratory data analysis (Fig 2), as well as the results of the convolutional model prior to class balancing (Fig 3).

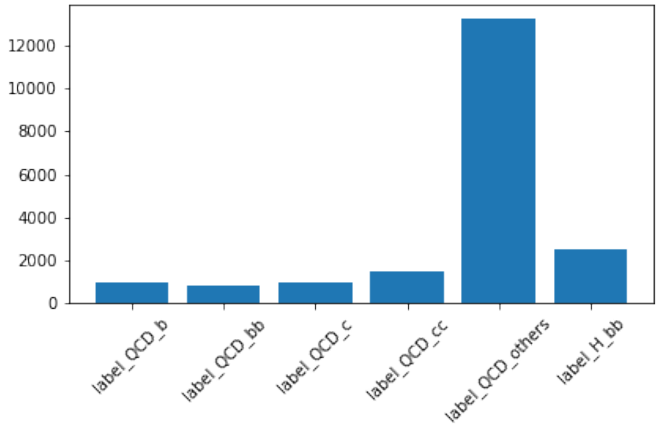

Figure 2. Distribution of class representation in our dataset

To correct this, balanced class weights were calculated following the formula below:

$$w_i = \frac{number\ of\ jets}{number\ of\ classes \cdot number\ of\ jets\ in\ class\ i}$$

With the inclusion of these weights applied to each class of particle jets that we concern ourselves with, the model greatly increases its performance (Fig 4).

In addition to modifying our previous model, we sought to investigate whether different model architectures would be more appropriate for our task. To more robustly determine the best architecture for multi classification on this dataset, we looked to other kinds of neural networks. We continued to tune our convolutional neural network (CNN) model, but upon further investigation, we decided to compare this model against an implementation of a graph neural network (GNN) [7].

The GNN model that we are extending to implement a multiclass classification feature is an interaction network to model the particle-particle interactions. The model takes in 48 track features and utilizes batch normalization layers to help stabilize the training. The GNN we implemented contains 3 update functions and 3 aggregation functions. Each update and aggregation function pair will make up the process of a single graph network (GN) block including: edge block, node block, and global block. The edge block is used to update the edge features from the input and receiver nodes. The node block is used to update the edges and the global block is used to set the output nodes.

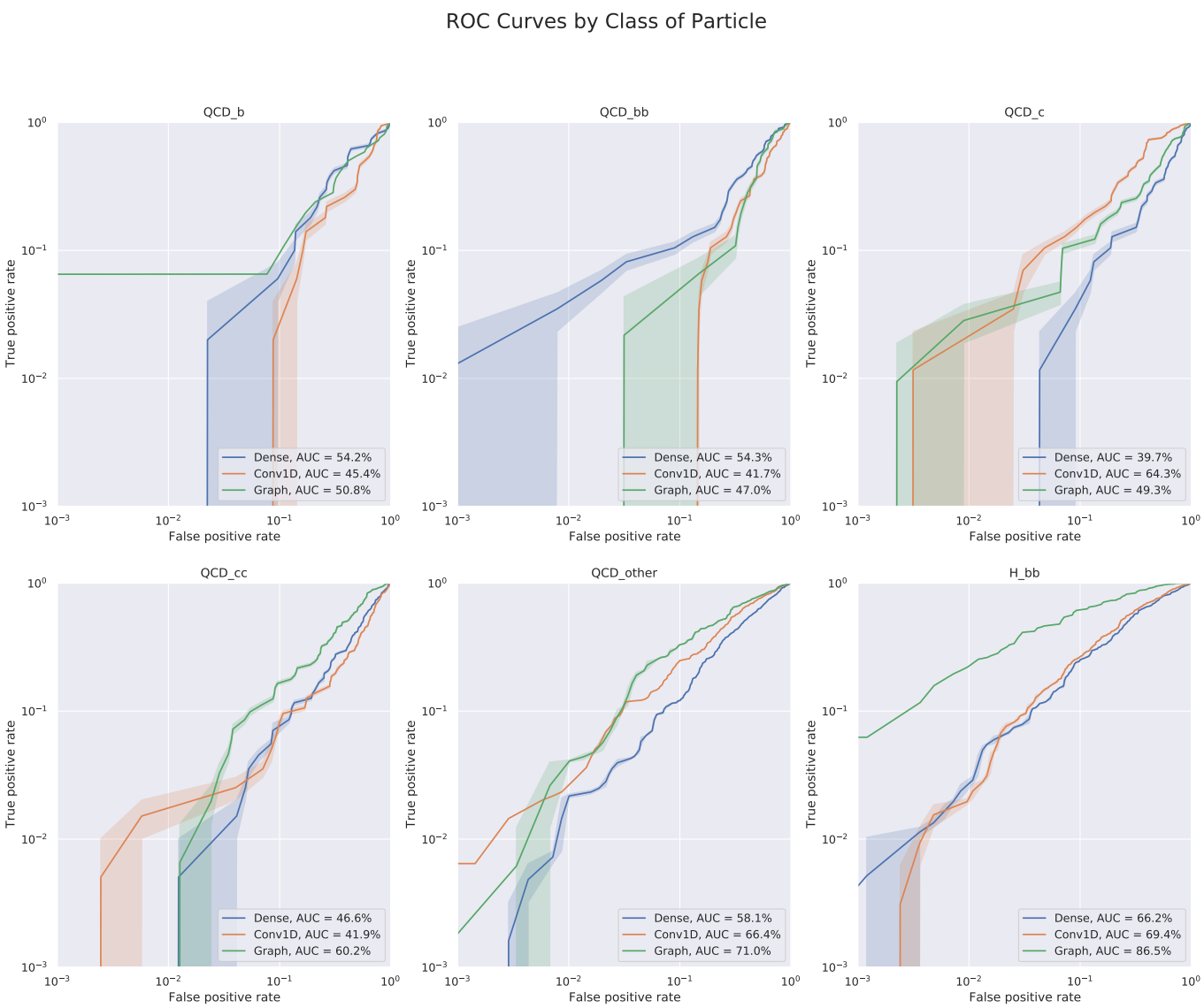

Figure 3. ROC Curves for models before class weighting

3 Results

We successfully added a multiclass classification feature to two deep learning models: 1 dimensional convolutional neural network (Conv1d) and graph neural network (GNN), producing predictions that classify each of the 6 different categories of jets. As a baseline when training the models on only one training file while also neglecting the skewed distribution of data, we can identify poor performances by the models.

For the Dense and Conv1d models, we noticed major improvements from previously unfavorable classification performances after feeding the models multiple training files and balancing the class weights. The largest improvement can be identified as the Higgs boson jet, improving from 86.5% to 96.5%.

Regarding the GNN, the baseline model without the jets performs fairly well without accounting for the class weights. Upon trying to improve its performance, using class weights proved to be difficult. Although weights were calculated identically for the Dense and Conv1d models, incorporating class weights in the GNN did not significantly affect performance. This feature is still being tested for improvement and will be resolved using more resources in the future.

4 Conclusion

Across the 3 models, the jet category with the most data had the best performances from all the models. For

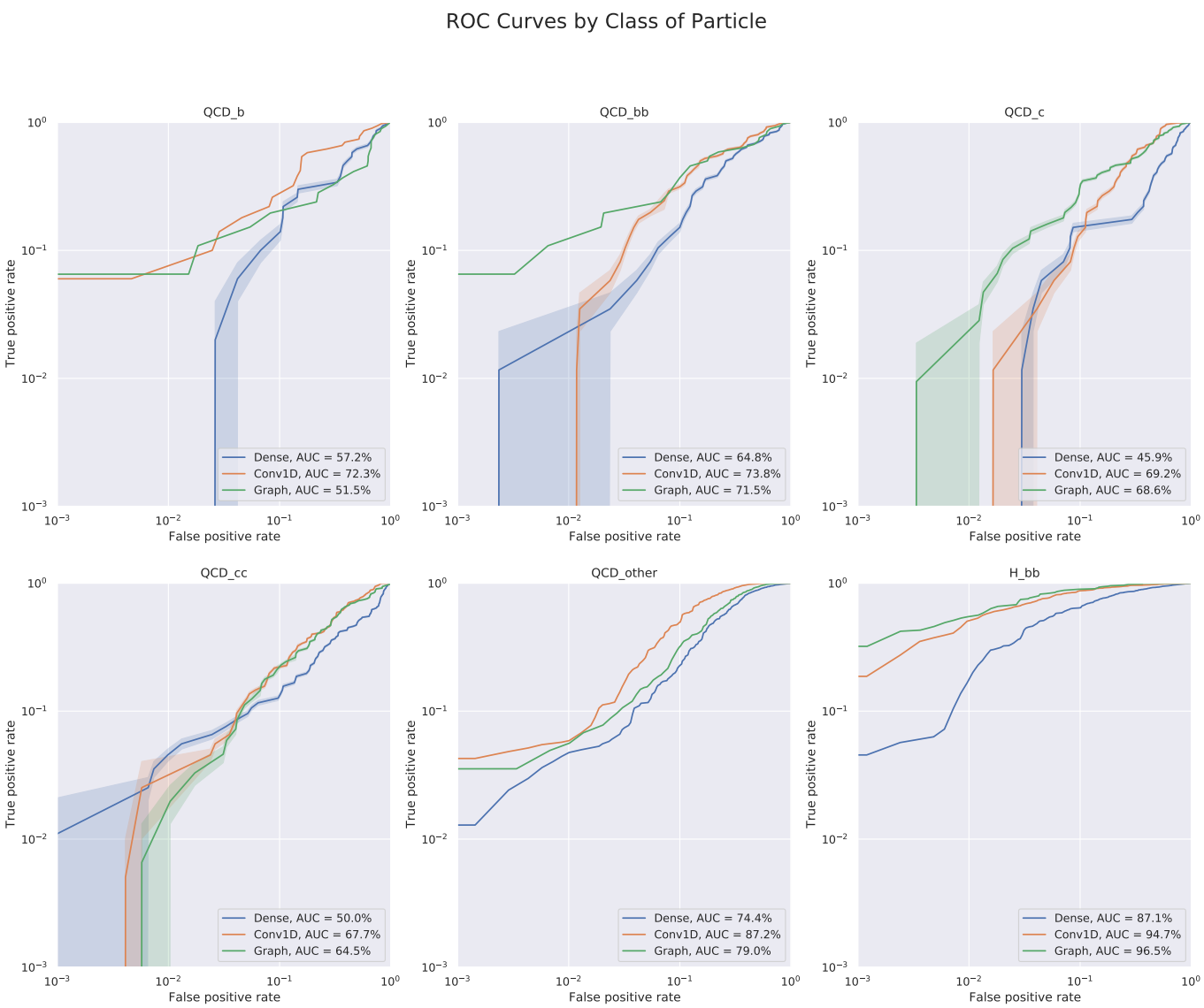

Figure 4. ROC Curves for models after class weighting

instance, the Higgs boson had an AUC of 96.5%, along with a higher percentage of data points compared to the others. This further shows the hurdle caused by the imbalance in data across the 6 jet classes. Although Higgs boson also didn't have too much representation in the data its unique quality that outshined the other jet classes is what we hypothesize causes us to be able to classify it with high accuracy.

Discriminating between the different jets is already a difficult task. While the Higgs boson has more distinguishable features, other jets have close similarities, like the two charmed quarks with the two bottom quarks and the single charmed quarks with the single bottom quarks. These minimal differences most likely contribute to the poor classification performances on top of the lack of data.

Understanding the inner workings of new neural networks was a major difficulty we faced. In order to debug the code for the GNN model, we required research and mentorship. Even so, some issues, such as the lack of significant improvement when implementing class weights for the model, were left unresolved for the time being.

In the future, we would have liked to implement an Equivariant Neural Network (ENN). ENNs are similar to graph networks, with the additional feature of respecting the symmetry in physics, a very important characteristic in particle jet classification. However, ENNs are not as widely used as CNNs or GNNs, resulting in fewer resources to reference for implementation. Given a larger time frame, we would like to compare a baseline ENN to

our existing models.

Our goal was to explore deep learning multiclassification techniques for classifying 6 different categories of particle jets, comparing several possible baseline models for jet tagging. By building a multi classifier, we simplify the process from creating individual classifiers for each jet to one large model. In doing so, we make an already tedious task more efficient. This project can be used as a stepping stone for future projects in the intricate world of particle physics.

Acknowledgements

We would like to thank our mentors Javier Duarte and Frank Wurthwein, our TA Farouk Mokhtar, our instructor Aaron Fraenkel, and our domain peers Alex Luo and Cecilia Xiao.

References

[1] E. A. Moreno, T. Q. Nguyen, J.-R. Vlimant, O. Cerri, H. B. Newman, A. Periwal, M. Spiropulu, J. M. Duarte, and M. Pierini, "Interaction networks for the identification of boosted h→bb¯ decays," *Physical Review D*, vol. 102, Jul 2020.

[2] J. Duarte, "Sample with jet, track and secondary vertex properties for Hbb tagging ML studies HiggsToBBNTuple_HiggsToBB_QCD_RunII_13TeV_MC," *CERN Open Data Portal*, 2019.

[3] P. T. Komiske, E. M. Metodiev, and J. Thaler, "Energy flow networks: deep sets for particle jets," *Journal of High Energy Physics*, vol. 2019, Jan 2019.

[4] H. Qu and L. Gouskos, "Jet tagging via particle clouds," *Physical Review D*, vol. 101, Mar 2020.

[5] J. Bjorck, C. P. Gomes, and B. Selman, "Understanding batch normalization," *CoRR*, vol. abs/1806.02375, 2018.

[6] A. F. Agarap, "Deep learning using rectified linear units (relu)," *CoRR*, vol. abs/1803.08375, 2018.

[7] J. Shlomi, P. Battaglia, and J.-R. Vlimant, "Graph neural networks in particle physics," *Machine Learning: Science and Technology*, vol. 2, Jan 2021.