Particle Physics Domain Checkpoint

Bryan Ambriz, Rui Lu, Charul Sharma

Halıcıoğlu Data Science Institute, University of California San Diego

DSC 180B: Capstone Project 2

Dr. Aaron Fraenkel, Dr. Javier Duarte

Introduction

Extending the content of last quarter of deep sets neural network, fully connected neural network classifier, adversarial deep set model and designed decorrelated tagger (DDT), we went a little bit further this quarter about picking up different layers in neural network like GENConv and EdgeConv. GENConv and EdgeConv play incredibly important roles here for boosting the performances of our basic GNN model. We also evaluated the performance of our model using ROC (Receiver-Operating Curve) curves describing AUC (Area Under the Curve). Meanwhile, based on previous experiences of project one and past project of particle physics domain, we decided to add one more section, exploratory data analysis in our project for conducting some basic theory, bootstrapping or common sense of our dataset.

But we have not produced all the optimal outcomes so far even though we finished the EdgeConv part and for the following weeks, we would like to finish the GENConv and may try some other layers to find out the potential to increase the performance of our model.

Motivations

After the fall quarter, we are still not satisfied with the performances of our models. Based on this, we decided to try several layers in this quarter to find out the optimal way for boosting the performance. And if it is possible, we will try to combine some methods in the fall quarter with the different layers together in a hybrid way.

Methods

GENConv

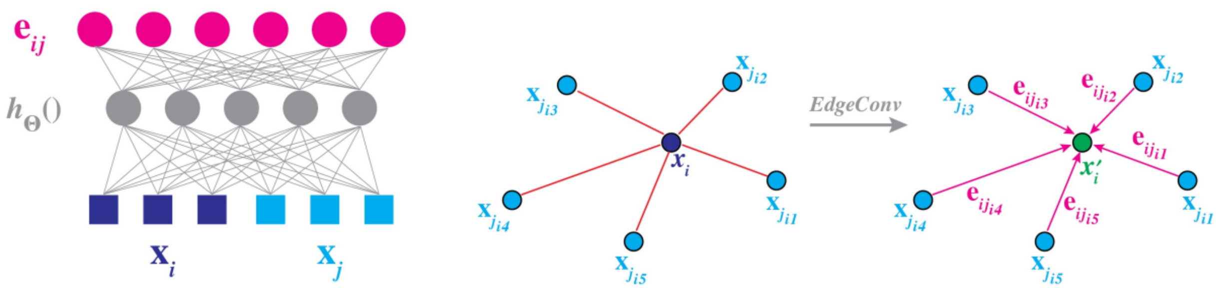
Basically speaking, GENConv is a model for implementation of GCNs(Graph Convolutional Networks). Regular GCNs model suffers from the vanishing gradient, overfitting and over-smoothing when the layers go deeper and deeperGCN here differentiate generalized aggregation functions like mean function and max function while the creators of it proposed MsgNorm as a new normalization layer and another pre-activation for GCNs. In other words, for exact operation and mechanism behind, the model combines information from the regular node and its neighbor by aggregating then updating the node feature by giving the aggregated value. As for regular GCNs, it just gives the information about connected neighbors for updating the node feature. Moreover, the specialty for the method is that the creators made a novel generalized aggregation function to cover all common sorts of aggregation functions. And according to their essay, the model boosted performance significantly.(Li, Xiong, Thabet, & Ghanem, 2020)

As for the results of GENConv so far, we have finished this and we plan to figure it out at the end of week 5.

EdgeConv

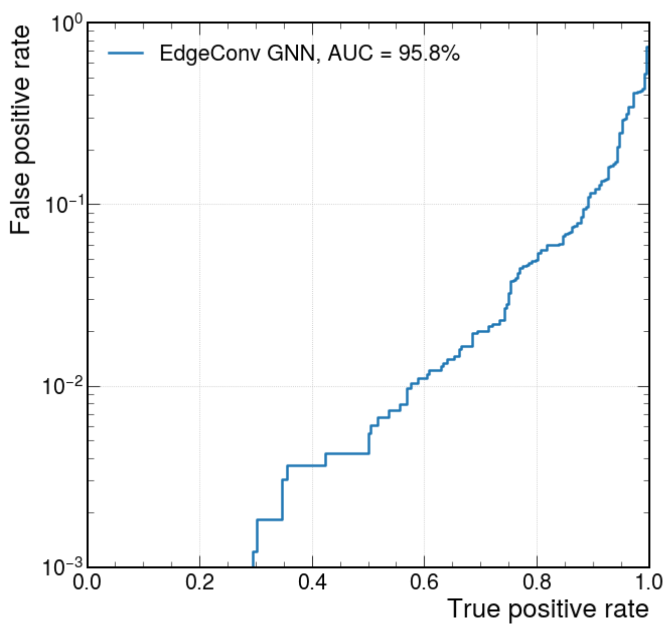
EdgeConv basically is the implementation of deep learning models which could tremendously increase the performance of these models because first, it acts on graphs dynamically computed in each layer of the model, incorporates local neighbor information and can be used for learning the global properties. Moreover, the EdgeConv helps the model to handle irregularity easily but the major difference between this and other methods is that it captures the local geometric structure while still maintaining its own permutation invariance. Besides that, EdgeConv could generate edge features that describe the relationship between a node and the neighbors and is capable of grouping nodes both in Euclidean space and in semantic space which would be extremely useful for particle physics.

(Wang et al., 2019)



Left: Computing an edge feature, ei j (top), from a point pair, xi and xj (bottom). In this example, hΘ() is instantiated using a fully connected layer, and the learnable parameters are its associated weights. Right: The EdgeConv operation. The output of EdgeConv is calculated by aggregating the edge features associated with all the edges emanating from each connected vertex.

As for the results of EdgeConv so far, the plot of AUC is all we have.



Measurement

As for measurement, we still choose to use the plots of AUC and the plots of ROC to represent the final performance of models. ROC(Receiver operating characteristic) here helps to measure the performance of our GNNs model at all classification thresholds using 2 parameters, True Positive Rate(TPR) and False Positive Rate(FPR).

Related Works & Discussion

To extend our model, we decided to use a graph neural network approach. In order to approximate the results of the paper we are expected to replicate, we aim to achieve or surpass the accuracy of the initial model we had utilized during quarter 1. Therefore, we have been conducting a literature review of graph neural networks which specifically utilize convolution, as we believe convolution and convolutional models can discover patterns not only between node features, but also between jets / i.e particle interactions. For example, we specifically looked at papers which go over EdgeConv (Wang et al., 2019) and GENConv (Li. et al., 2020) convolutional layers and their implementation in graph neural networks. At this point in time, we have successfully produced classification models that implement the EdgeConv convolutional layer.

Goals

Due to the approaching of the ending of winter quarter, we will try our best to test more layers and to see if it is possible to use DDT as an implement role for ROC because it is based not only on statistical discrimination power but also the robustness of inherent QCD background while it can be used for any heavy boosted objects.

Moreover, as the final presentation of our projects, we are still preparing the content of our website with specific columns for all kinds of audiences like people who are not familiar with particle physics, GNNs models or machine learning. Because the project should be readable for all audiences, we planned to add some columns like introduction of Higgs boson particle, explanations of GNN models and those layers like EdgeConv and GENConv used.

Reference

Li, G., Xiong, C., Thabet, A., & Ghanem, B. (2020, June 13). DeeperGCN: All you need to train deeper gcns. Retrieved February 04, 2022, from https://arxiv.org/abs/2006.07739

Wang, Y., Sun, Y., Liu, Z., Sarma, S., Bronstein, M., & Solomon, J. (2019, June 11). Dynamic graph CNN for learning on point clouds. Retrieved February 04, 2022, from https://arxiv.org/abs/1801.07829