High-Energy Particle Classification with Graph Neural Networks

Competition: High-Energy Particle Classification

Rank: 2nd Place

Team: mr_sandman

Name: Raphael Kiminya

Organization: Strathmore University

Abstract

Jets in particle physics are essentially point clouds with no apparent order that can be used to differentiate jets of one class from another. Conventional machine learning algorithms require a fixed number of inputs and therefore require summarizing the particle attributes to work on this kind of data. This in turn leads to information loss since the summaries are not as expressive as the raw data. Graph Neural Networks solve this problem by allowing the representation of a jet a graph, thereby enabling modeling on the raw particle data.

Key Words

graph neural network, particle cloud, edge convolution

Baseline

Early on I realized that jets in the same event have the same label. I therefore reframed the task to predict event-level labels based on jet and particle properties, then later assign jets in the same event the event label.

My baseline model was a gradient boosting model based on the summary properties of jets and particles. I calculated summary statistics such as mean, max, min, trend of particle/jet positions (x,y,z), energy, mass and category, then fitted a LightGBM model. At best, this model achieved local cross-validation accuracy and ROC AUC score of ~60% and ~0.7 respectively.

Summarizing the data was not ideal as it led to some information loss, meaning the summarized data could not be as expressive as the raw values. A more appropriate representation was needed.

Graph Neural Networks

While researching a more natural way to represent particles, I came across a paper by Isaac Henrion et al. (2017) [1]. The authors propose the representation of jets as a graph, with particles as nodes and node features derived from the 4-momenta of these particles. They then discuss several message-passing neural network (MPNN) designs and compare their performance on a binary classification task. In their experiments, the MPNN was able to achieve a ROC AUC score of ~0.92 in the classification of jets arising from quarks and gluons versus jets arising from *W* bosons.

Inspired by these results, I attempted to model the problem as a graph neural network (GNN) using PyTorch Geometric[1], a geometric deep learning extension library for PyTorch. The library contains implementations of a variety of published works on graph neural networks and provides a simple interface to work with them, making it easy to quickly prototype and test out ideas. I generated a particle graph dataset where each event formed a single graph with the particles as nodes. The raw particle attributes were formed the node features, with the particle particle_px, particle_py, particle_pz attributes doubling as node coordinates.

---

[1] PyTorch Geometric: https://github.com/rusty1s/pytorch_geometric

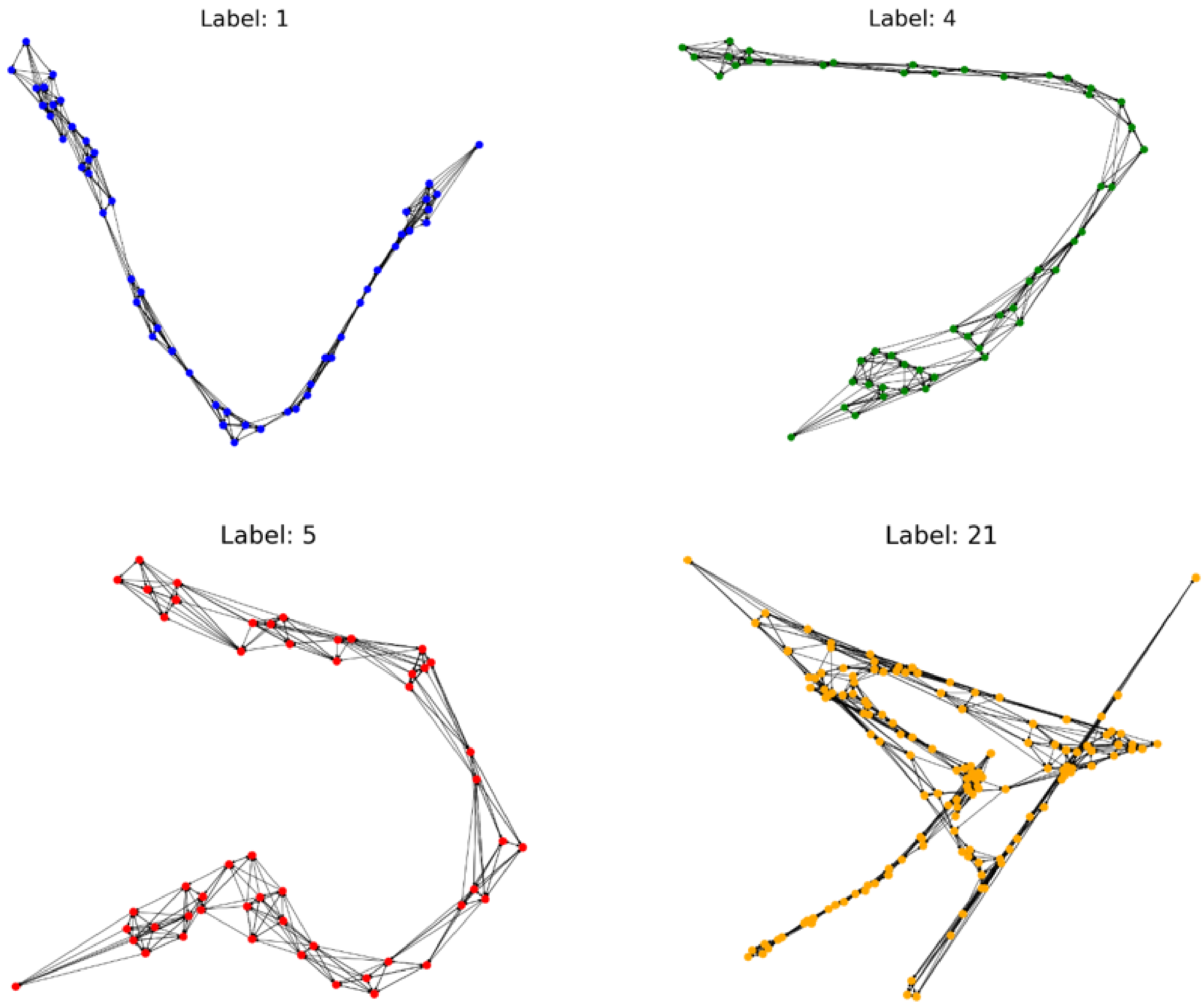

Figure 1: Graph visualization of the various event classes

I tried out a number of GNN modules provided in the library and found that the DynamicEdgeConv module with mean pooling produced the best results. The module is based on Yue Wang et al. (2018) [2], where the authors propose edge convolution (EdgeConv), a CNN-based neural network adapted for classification and segmentation tasks on point clouds. The graph edges are dynamically constructed using the *k*-nearest nearest neighbors in the feature space. The model performed best within the range of 12 – 16 neighbors. The complete set of parameters used is summarized in Table 1.

Component	Parameters
Node Features	particle_px, particle_py, particle_pz, particle_energy, particle_mass, particle_category
Node position	particle_px, particle_py, particle_pz
Network	4x stacked DynamicEdgeConv with *k*=15 neighbors and mean aggregation
Global Pooling	global mean pool

Table 1: PyTorch Geometric model architecture summary

After training for 20 epochs, the best model achieved a local validation accuracy of 75% and ROC score of 0.83 (0.81 on the competition leaderboard).

ParticleNet

Point cloud representation and GNNs was clearly the way to go so I focused my research on the topic. I came across Qu and Gouskos (2019) [3], where the authors discuss various jet representations and propose particle clouds as the more natural representation. They then introduce the ParticleNet architecture, a CNN-like deep neural network for jet tagging with particle cloud data. ParticleNet is essentially a stack of EdgeConv blocks with a global pooling operation.

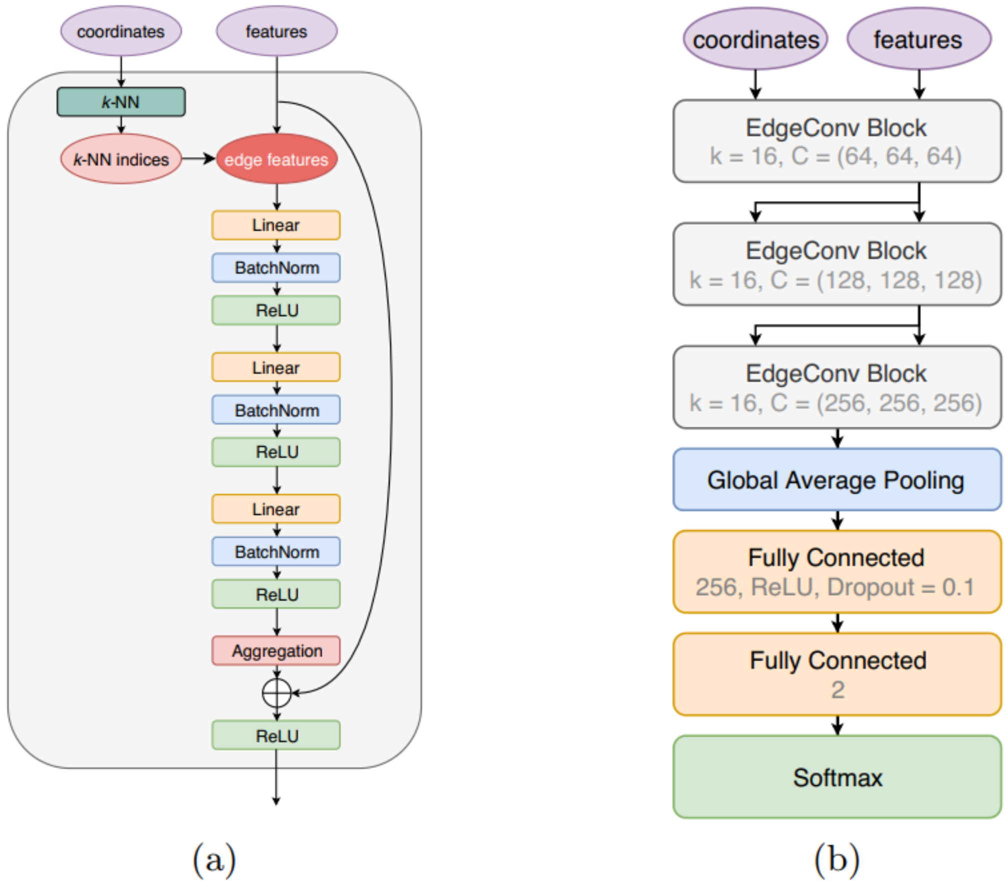

Figure 2: The architecture of (a) the EdgeConv block, (b) the ParticleNet network

I based my final model on Huilin Qu's TensorFlow implementation of ParticleNet [2] with modified input features and training parameters. The implementation also includes extra features derived from the raw features. An ensemble of 4 models trained on different parts of the dataset with different slightly different training parameters achieved a leaderboard score of 0.86.

[2] ParticleNet: https://github.com/hqucms/ParticleNet

References

[1] Isaac Henrion et al. “Neural message passing for jet physics”. In: (2017). URL: https://dl4physicalsciences.github.io/files/nips_dlps_2017_29.pdf

[2] Yue Wang et al. “Dynamic Graph CNN for Learning on Point Clouds”. In (2018). arXiv:1801.07829v2 [cs.CV]

[3] Huilin Qu and Loukas Gouskos. “ParticleNet: Jet Tagging via Particle Clouds”. In: (2019). arXiv: 1902.08570 [hep-ph]