GNN (IceCube 2D)

4/21/2025

3 Möjliga Poäng

Försök 1

4/11/2025

NÄSTA: Feedback från granskning

Poäng för försök 1: Ej tillämpbart

Lägg till kommentar

Obegränsat antal försök tillåts

4/22/2025

Information

In this exercise, you will implement a Dynamic Edge Convolution Graph Neural Network to reconstruct the position of the neutrino interaction from the

Please see **IceCube dataset** (https://uppsala.instructure.com/courses/102453/pages/icecube-dataset) and **gnn_neutrino_dataset_description.pdf** (htt 7f4d9440bbed%2Fgnn_neutrino_dataset_description.pdf%3Ftoken%3DeyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJpYXQiOjE3NDQ5NjQ0NTEsInVzZXJfa for a description of the dataset. Every event consists of a list of observed photons (aka hits or pulses). Each photon is described by its detection time t

We recommend using the **PyTorch Geometric** (https://pytorch-geometric.readthedocs.io/en/latest/index.html) package and its **Dynamic EdgeConv** neighbors from the feature space, resulting in k pairs of nodes. Each pair is combined into a single array of size twice the feature size, as illustrated in

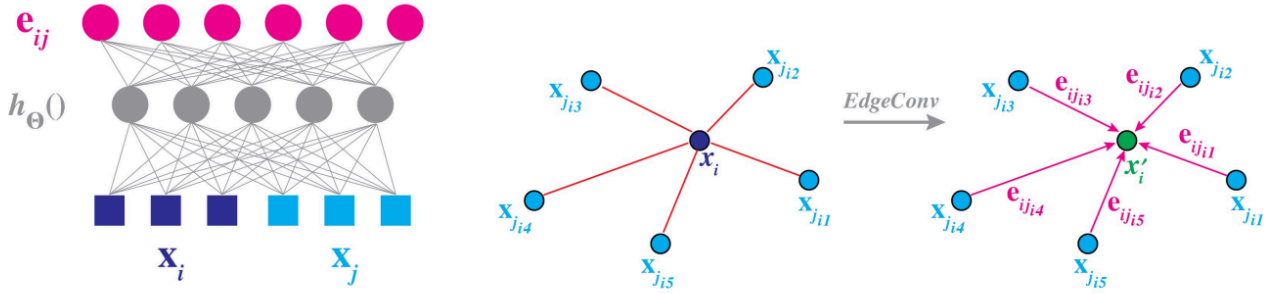

Fig. 2. **Left**: Computing an edge feature, $\mathbf{e}_{ij}$ (top), from a point pair, $\mathbf{x}_i$ and $\mathbf{x}_j$ (bottom). In this example, $h_\Theta()$ is instantiated using a fully connected layer, and the learnable parameters are its associated weights. **Right**: The EdgeConv operation. The output of EdgeConv is calculated by aggregating the edge features associated with all the edges emanating from each connected vertex.

Your task is to implement a Graph Neural Network using several (at least two) Dynamic Edge Convolution layers to predict the position of the neutrino

This template might be useful to get started: **train_gnn_template.py** (https://uppsala.instructure.com/courses/102453/files/8368182?wrap=1) (https://

Evaluate the performance by calculating relevant metrics and visualizing the results in appropriate plots.

Remember always to include:

- A **written summary** (0.5–1 A4 page) covering (submitted either as PDF or directly as text):
  - What you did and how
  - What results you obtained
  - What challenges you encountered and what could be improved
- A **PDF (or similar format)** with all **result plots**, each with a short explanation
- Your **code**, preferably as a link (e.g., GitHub, Google Colab, etc.) so we can view it easily.

Installation hints:

- You need to install torch_geometric and torch_cluster. See **https://pytorch-geometric.readthedocs.io/en/latest/install/installation.html** (http
- Please note that the newest PyTorch version, 2.6, is not yet supported. To avoid incompatibilities, please use PyTorch 2.5.
- Apple Silicon (MPS) is not supported (but it should still train reasonably fast on CPU, I get ~10 seconds per epoch on an M2 Pro)
- You can use the "selector" on **https://pytorch-geometric.readthedocs.io/en/latest/install/installation.htm** (https://pytorch-geometric.readthed
- On Google Colab you need to downgrade to PyTorch2.5 and then install torch_geometric and torch-cluster via
  - !pip install torch==2.5
  - !pip install torch_geometric
  - !pip install torch-cluster -f https://data.pyg.org/whl/torch-2.5.0+cu124.html
  - A restart of the runtime might be required before the installation has effect

For those of you who want to test your skills on a realistic dataset, you can check out this Kaggle challenge of the IceCube Neutrino Observatory: **http**

Filnamn	Storlek	
Exercise_...k (1).pdf	1,73 MB	✓

	Filnamn	Storlek	
	train_gnn_template.py	3,7 kB	
	gnn_encoder.py	5,29 kB	
	gnn_trafo_helper.py	10,5 kB	

(https://uppsala.instructure.com/courses/102453/modules/items/1326640)

(https://uppsala.instructure.com/courses/102453/modules/items/1346763)