

Explanation of the GraphSAGE Node Classification Code

Name : Sarah ibrahim kadas (2205223)

1) Setup

Installs PyTorch Geometric.

Imports:

- Data represents a graph (nodes, edges, labels).
- SAGEConv GraphSAGE convolution layer.
- Functional API (F) for activations and loss.

2) Defining Node Features

Each node has two features

- [1, 0] benign
- [0, 1] malicious
- Total nodes: 6.

3) Creating the Graph Structure

Graph is undirected, so each edge is added twice.

Structure:

Nodes 0,1,2 are fully connected (benign cluster).

Nodes 3,4,5 are fully connected (malicious cluster).

A bridge edge links node 2 \rightarrow 3 (interaction between benign & malicious groups).

4) Adding Labels

Labels:

- 0 = benign
- 1 = malicious

5) Building the PyG Data Object

- $x \rightarrow$ node features
- `edge_index` \rightarrow connectivity
- $y \rightarrow$ labels

6) Defining the GraphSAGE Model

Two GraphSAGE layers:

- `conv1`: learns neighbor-aware node embeddings.
- `conv2`: outputs final class scores.

Activation: ReLU

Final output: log-softmax (for classification)

7) Training the Model

- Input dimension: 2 features per node
- Hidden layer: 4 units
- Output classes: 2 (benign/malicious)
- Loss function: Negative Log-Likelihood, matching `log_softmax`
- Training runs for 50 epochs

The model learns based on:

- Node features
- Graph connectivity
- Label supervision

8) Generating Predictions

The model predicts a label (0 or 1) for each of the 6 nodes.