

# Simple GCN Code Explanation

## 1. Installing Libraries

First, we install **torch\_geometric**, which is a library for Graph Neural Networks. Then we import:

- **torch** → the main PyTorch library
- **GCNConv** → the graph convolution layer
- **Data** → used to store the graph (nodes + edges)

## 2. Building the GCN Model

We create a class called **GCN**.

Inside the model we have:

- **conv1**: the first graph convolution layer
- **conv2**: the second graph convolution layer

In the **forward** function:

1. We take **x** (the features of each node).
2. We take **edge\_index** (the connections between nodes).
3. We pass them through the first layer.
4. We apply ReLU activation.
5. We pass the result through the second layer.
6. We return the final output.

## 3. Creating the Graph

- **x** contains the feature of each node (1 to 5).
- **edge\_index** shows which nodes are connected.
- **Data(x, edge\_index)** puts the node features and edges together in one graph object.

## 4. Running the Model

We create a model with:

- **1 input feature**
- **4 hidden features**
- **2 output features**

Then we run the graph through the model.

The output has:

- **5 rows** → one for each node
- **2 columns** → the output features of each node

### Final Output

The output numbers change every time because the model starts with random weights.  
But the shape is **always**  $5 \times 2$ .