For Those Who Don't Know the Jargon, Data Enthusiasts Welcome!

What exactly is Graph Neural Networks (GNN) ?

Sanjay N Kumar

Data scientist | AI ML Engineer | Statistician | Analytics Consultant

What Is a Graph? 🧩

A **graph** is a way to show **connections**.

🟢 **Nodes (Vertices)** = Items (e.g., people, cities, proteins)

🔗 **Edges (Links)** = Relationships between items (e.g., friendships, roads, chemical bonds)

📍 **Real-Life Examples**:

- 👨‍👩‍👧‍👦 **Facebook:** People are nodes, friendships are edges
- 🗺️ **Maps:** Cities are nodes, roads are edges
- 🧬 **Molecules:** Atoms are nodes, bonds are edges

What Is a Graph Neural Network (GNN)? 🔍

GNN is a **deep learning model** that works on **graphs**.

Unlike regular neural networks (which work on tables or images), GNNs learn from **connections**.

📌 **Key Idea:**

A node's new information = its own info + neighbors' info

📊 It captures the **structure** of the data

🧠 GNN helps machines think more like humans – ***"Who are your friends?"*** 👥

How GNN Works - In 4 Steps 🧠

Let’s break it down 🔄

1️⃣ **Message Passing**: Each node sends messages to its neighbors

2️⃣ **Aggregation**: Collect messages from all neighbors

3️⃣ **Update**: Combine old info with new info

4️⃣ **Repeat**: Do this for multiple layers/rounds

📦 It’s like learning not just from your own homework, but from your friends’ too!

Step 1 - Message Passing 📨

What Happens?

Each node **sends a message** to its neighbors.

📘 Imagine you're in a class 👩‍🏫. You tell your friends what you know about a subject.

Each friend also shares what *they* know. 🗣️

📊 **In GNN:**

Each node **shares its data (features)** with all connected nodes.

Step 1 - Message Passing 📨

🧪 **Example:**

Node A = 2, Node B = 5 → A sends 2 to B, and B sends 5 to A.

🧠 This builds a network of shared knowledge!

Step 2 - Aggregation 🧮

What Happens?

Each node **collects** the messages it got from its neighbors and **merges** them.

📘 Like when your teacher collects everyone's ideas and combines them into one summary 💡.

📊 In GNN:

Use math like **sum, mean, or max** to combine the values.

Step 2 - Aggregation 🧮

🧪 **Example:**

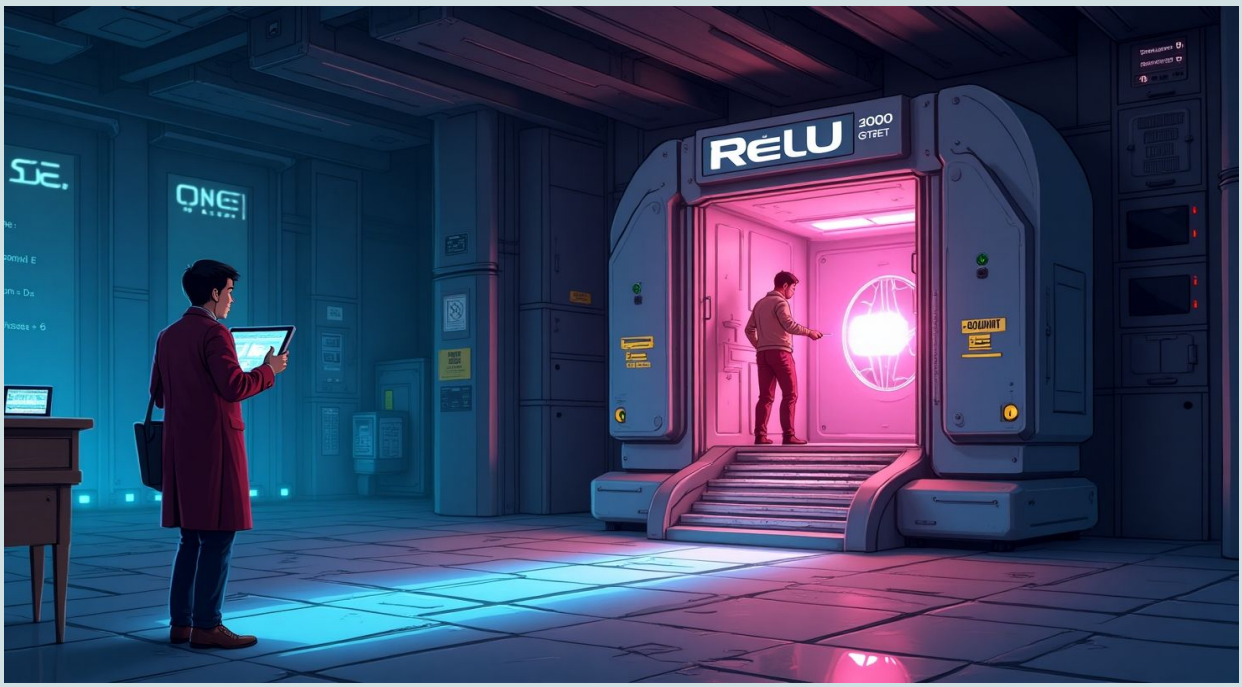
If Node A receives 5 and 7 from its neighbors,

ini

Aggregation = (5 + 7) / 2 = 6

💡 Node A now has a sense of what its neighborhood is like.

Step 3 - Update 🔁



What Happens?

Each node uses the aggregated message to **update** its own feature.

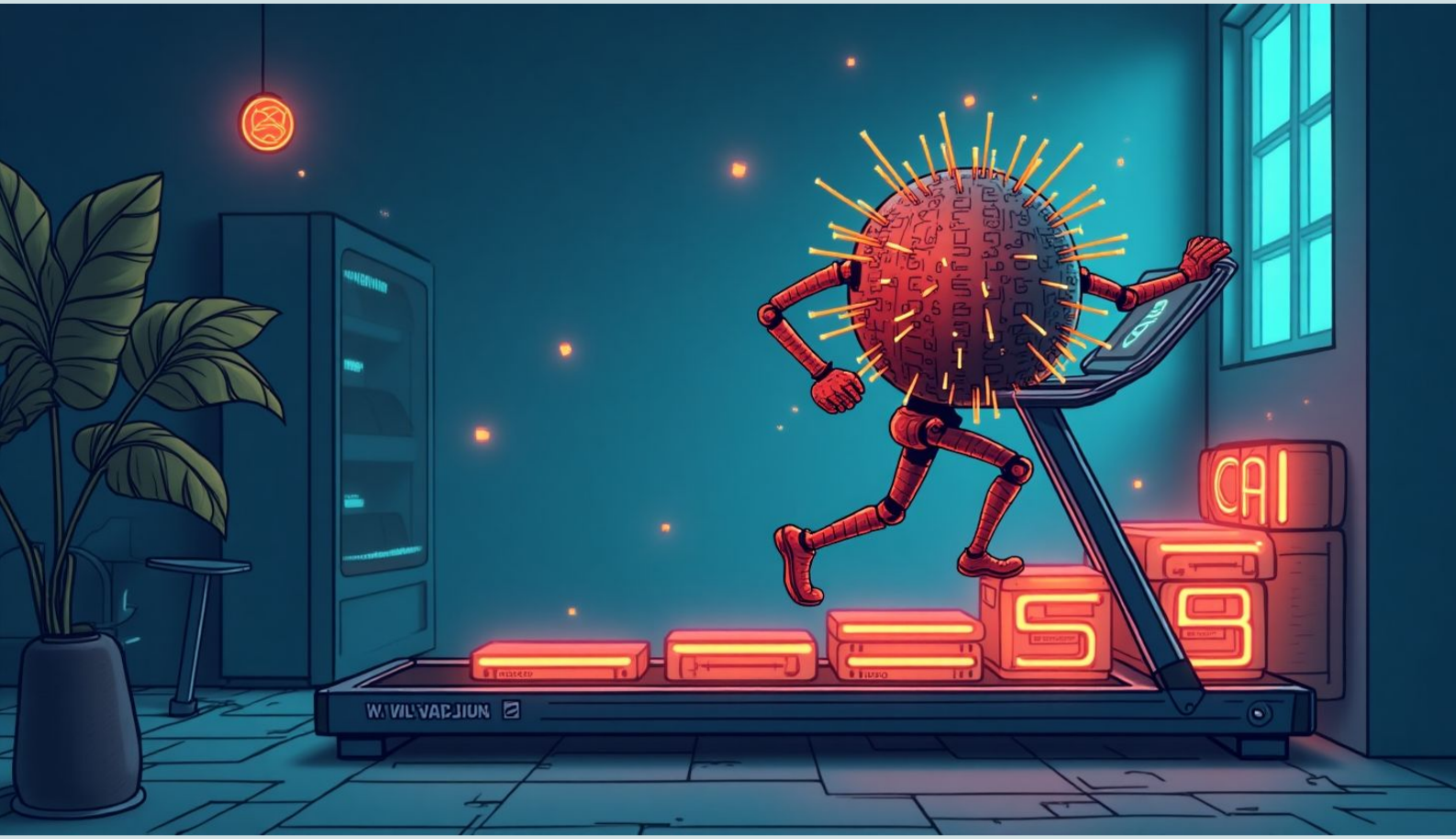
📘 Like you learning something new after hearing your friends' thoughts. 🧠

You *mix* your original idea with their ideas to get a better one.

📊 **In GNN:**

Use a small neural network layer (like linear + ReLU) to update the node's features.

Step 3 - Update 🔁



🧪 **Example:**

Original value = 2

Aggregated value = 6

Updated value = **some function of (2, 6)**, maybe:

$$\text{Updated} = \text{ReLU}(W1 \times 2 + W2 \times 6)$$

This helps each node learn a **better representation**.

Step 4 - Repeat 🔁🔁🔁

What Happens?

The GNN repeats the process for **multiple rounds** (called layers).

📘 It’s like discussing with friends again tomorrow — now you all know more! 🤝

Each round allows you to learn from friends of friends.

📊 **In GNN:**

- **1 layer =** info from direct neighbors
- **2 layers =** info from neighbors of neighbors
- **3 layers =** info from even further out!

💥 More rounds = **wider understanding** of the graph.

Simple Math Behind GNN 🧮

Let's say node A has two neighbors: B = 5, C = 7
Node A's value = average of neighbors

New_A = (B + C)/2 = (5 + 7)/2 = 6

🧠 Now, A learns the *context* from its neighbors.

You can also use:

- **Sum**
- **Max**
- **Weighted Average (if some friends are more important)**

More Real-Life Examples 🏠

🛒 **E-commerce**: Products bought together → Graph

🚕 **Traffic**: Roads as nodes, connections as edges

🎬 **Netflix**: Users & movies connected by watch history

🧪 **Chemistry**: Atoms in molecules form a graph

🔐 **Banking**: Detecting fraud by checking transaction patterns

Difference from Other Neural Networks 🔁

- **CNN (Convolutional Neural Network)** 🖼️

Used for images. It learns by looking at nearby pixels — like how your eyes notice patterns in a photo.

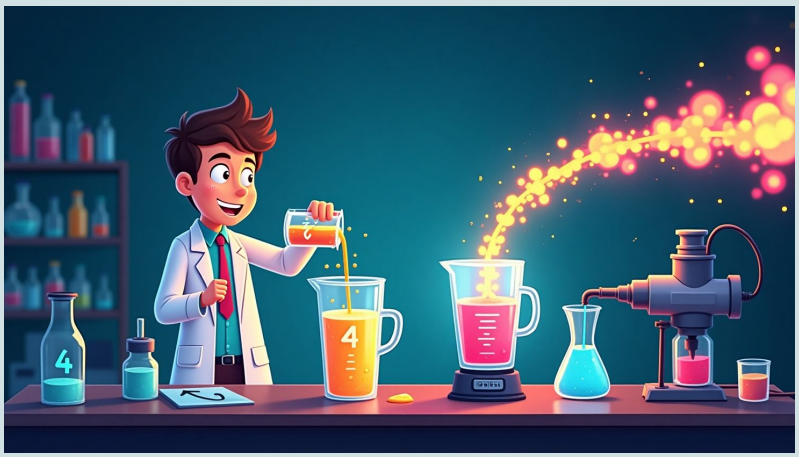
- **RNN (Recurrent Neural Network)** 📜

Used for sequences like text or time series. It learns from the past — each step depends on the one before it.

- **GNN (Graph Neural Network)** 🌐

Used for graphs. It learns from connected neighbors — like how you learn from your friends and their friends.

GNN Layer Example 🧱



📌 Each layer updates a node like this:

h_i = f(h_i, Aggregate(h_j for all neighbors j))

Let’s say:

- Node has value 2
- Two neighbors: 4 and 6
- Aggregation = mean
- Then:

New value = f(2, mean(4,6)) = f(2, 5)

Where f is a simple neural network layer (e.g., linear + ReLU)

GNN Use Case - Predict Disease Spread 🧪

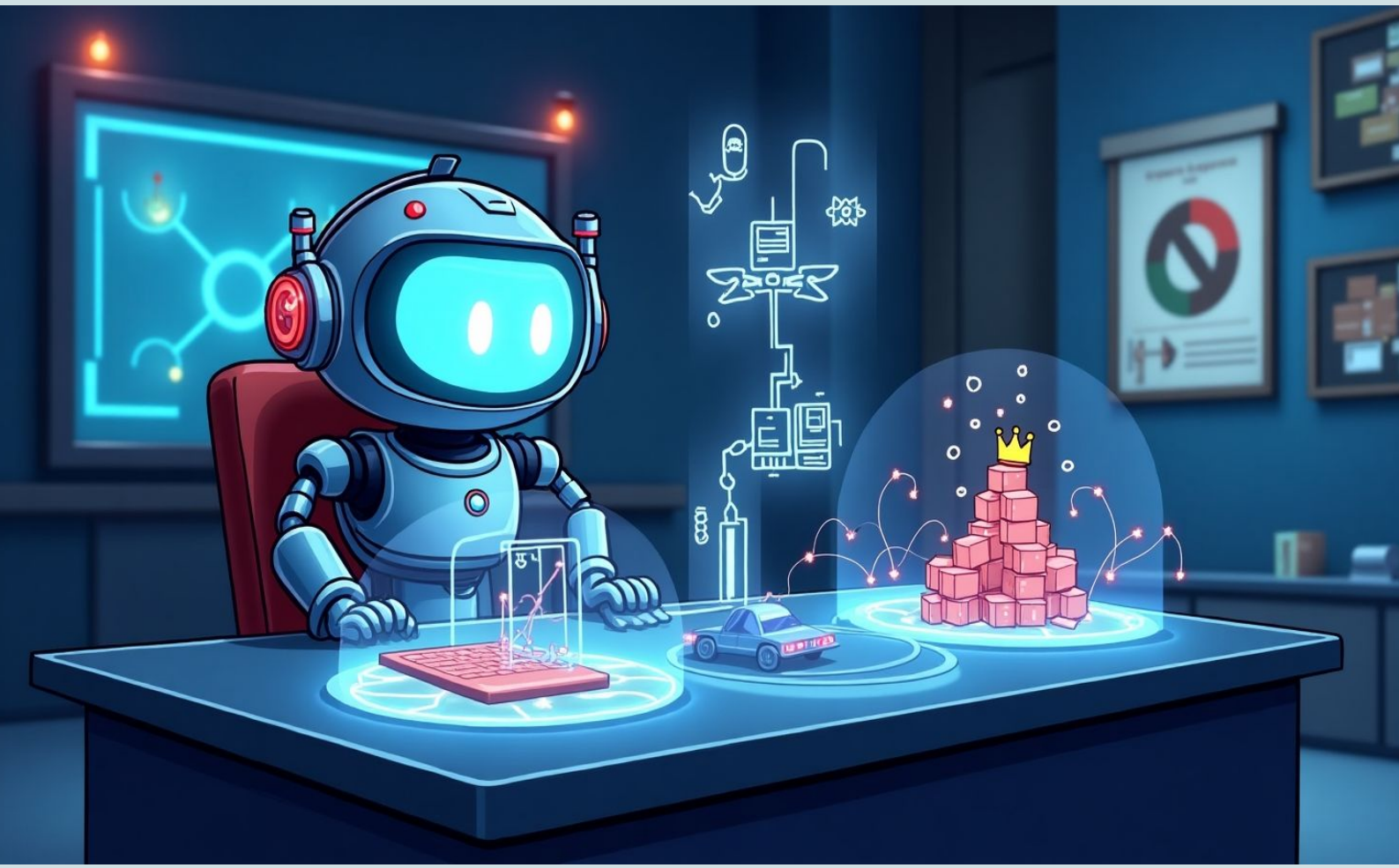
😷 A person is infected.

🔗 GNN sees who they met (nodes + edges).

🧠 It learns the pattern of spread to **predict the next possible infected person**.

Used in **COVID models**, **network biology**, etc.

Real-World Use Cases 📈



✅ **Social Networks** – Who might you know?
✅ **Search Engines** – Which page is important?
✅ **Drug Discovery** – Which molecule works better?
✅ **Traffic Flow Prediction** – Predicting congestion
✅ **Fraud Detection** – Suspicious connections in financial transactions

GNN vs Traditional Neural Network

📐

🏛️ **Traditional NN:** Good for tables/images

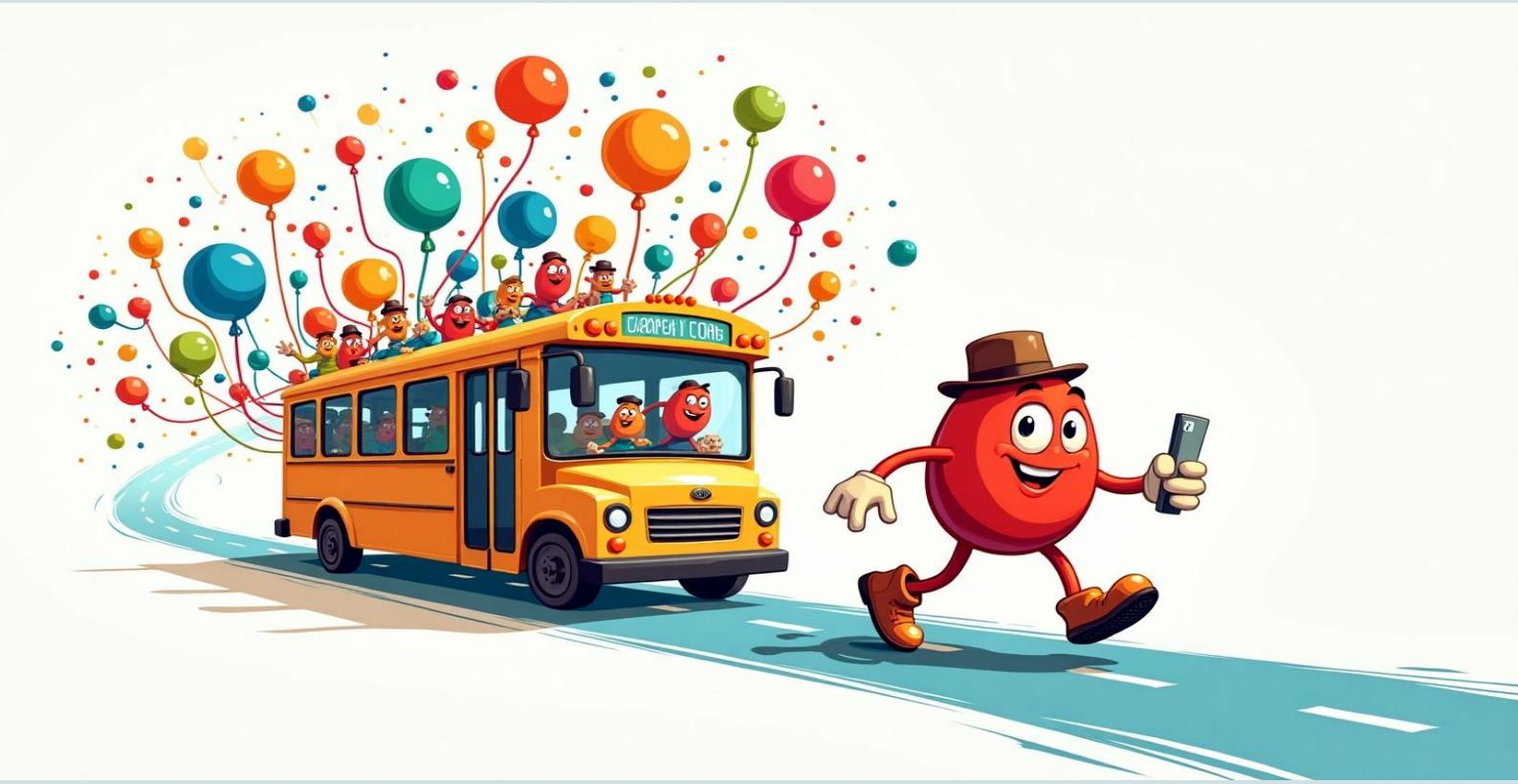
🌐 **GNN:** Good for networks and connections

📍 If your data involves *who is connected to whom*, GNN is the best choice.

Types of GNNs (For Curious Readers)

- **GCN (Graph Convolutional Network)** – Basic model
- **GAT (Graph Attention Network)** – Gives importance to certain neighbors
- **GraphSAGE** – Learns from a *sample* of neighbors
- **PinSage** – Used by Pinterest for recommendations!

Summary ❤️



- 🟢 **Graph =** Nodes + Edges
- 🧠 **GNN =** Learns by looking at **neighbors**
- 📊 Used in social media, science, maps, fraud detection
- ➕ Math is simple (Average or Sum of neighbors)
- **GNN =** Powerful for **networked data**

Decode Connections with GNN 🔑💡

From friends to formulas, GNN sees what others miss.

Let’s turn networks into knowledge — one node at a time. 🤝📊

📬 Reach out — and let’s build smarter systems, one connection at a time.

Sanjay N Kumar

Data scientist | AI ML Engineer | Statistician | Analytics Consultant

https://www.linkedin.com/in/sanjaytheanalyst360/

sanjaytheanalyst360@gmail.com