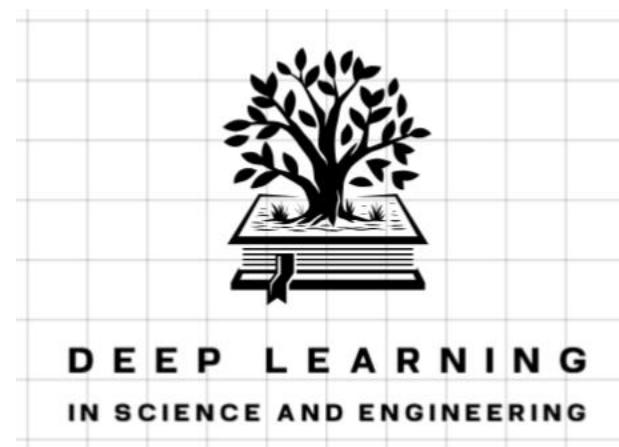


Graph Neural Network



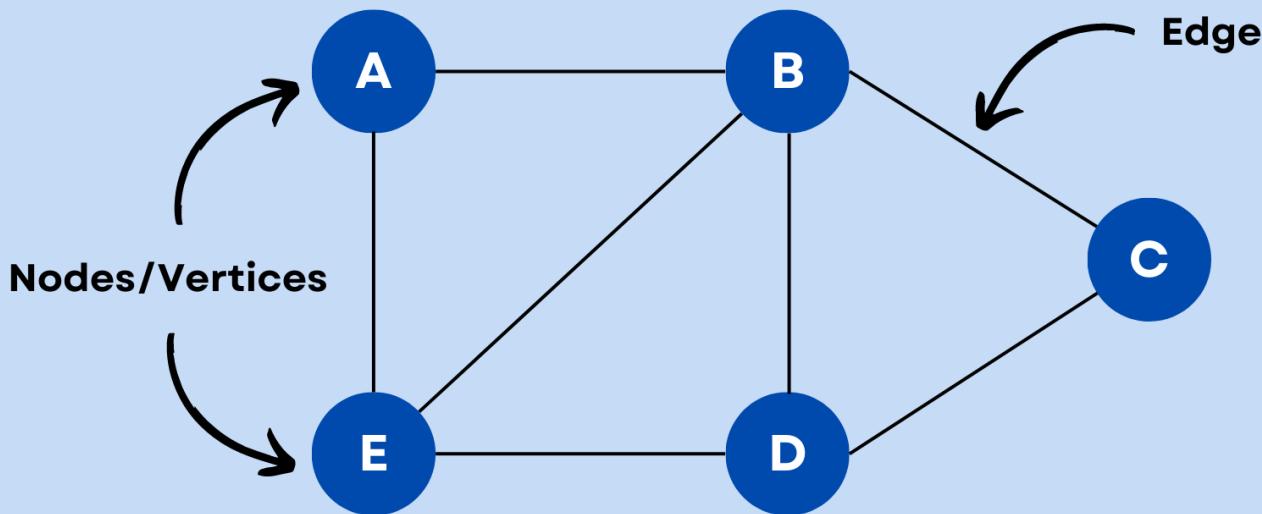
圖像 (Image)

- 定義：指照片、插畫、數位影像等用來呈現視覺訊息的圖形資料。
- 用途：用於視覺化展示，通常包含藝術性或紀錄性內容，例如風景照片、肖像畫等。
- 範例：
 - 人物的照片是一種圖像。
 - 用於產品設計的草圖也是一種圖像。

圖表 (Graph)

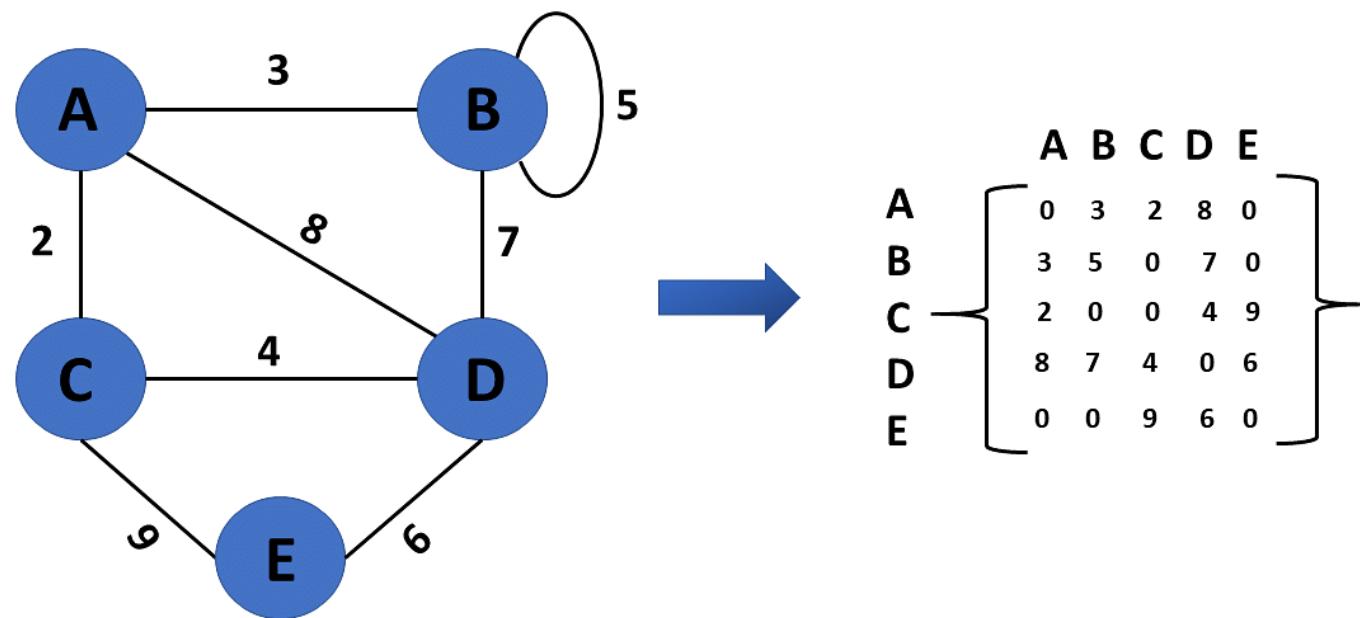
- 定義：指用數據或資訊視覺化呈現的工具，常用於解釋關係或趨勢，如折線圖、柱狀圖等。
- 用途：用於分析數據或展示邏輯關係，通常應用於研究報告、商業簡報等情境。
- 範例：
 - 用於展示銷售趨勢的折線圖是圖表。
 - 說明流程的流程圖也是一種圖表。

Graph Structure

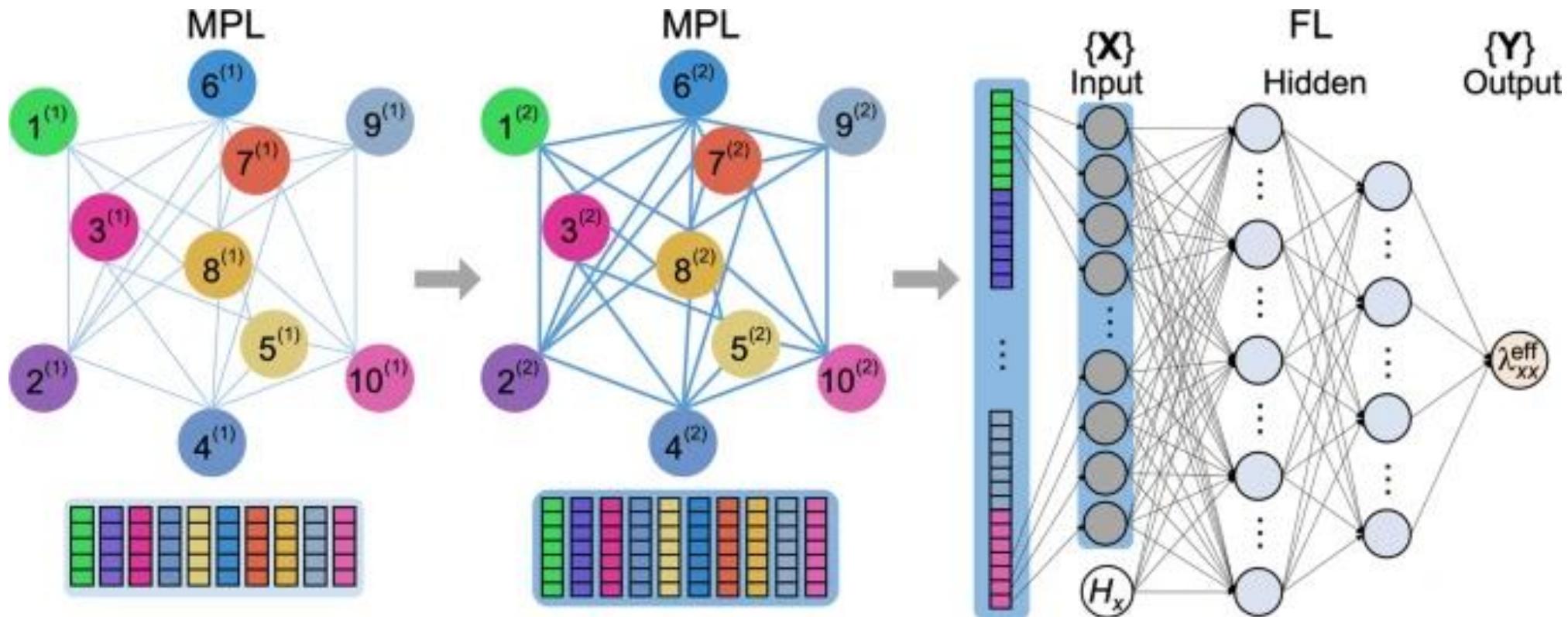


BOARD

Graph Structure



Graph Neural Network



圖神經網絡（ Graph Neural Networks，簡稱 GNN ）是一類專門處理圖結構數據的神經網絡，通過在圖中進行信息的傳遞與聚合，來學習節點和邊的表示。GNN 的主要優勢是能夠捕捉圖的結構信息，因此被廣泛應用於社交網絡分析、推薦系統、知識圖譜、分子結構預測等領域。

1. 圖結構的基本概念

圖 $G = (V, E)$ 由以下部分組成：

- 節點集 V ，其中每個節點代表一個實體，通常用 v_i 表示節點 i 。
- 邊集 E ，其中每條邊 (i, j) 表示節點 i 與節點 j 之間的關聯或交互。
- 節點特徵矩陣 X ，其中每行是節點的特徵向量。
- 鄰接矩陣 A ，該矩陣描述了圖中節點之間的連接關係，若 $A_{ij} = 1$ 表示節點 i 和節點 j 之間有邊，否則為 0。

2. 信息傳遞與聚合

GNN 的核心思想是每個節點通過聚合其鄰居節點的信息來更新自己的表示。這一過程通常可以分為兩個步驟：信息聚合和特徵更新。

a. 信息聚合

每個節點 v_i 通過其鄰居節點來聚合信息。對於節點 i ，其聚合信息可以表示為：

$$m_i^{(k)} = \text{AGGREGATE} \left(\{h_j^{(k-1)} : j \in \mathcal{N}(i)\} \right)$$

其中：

- $h_j^{(k-1)}$ 是節點 j 在第 $k - 1$ 層的特徵。
- $\mathcal{N}(i)$ 是節點 i 的鄰居節點集合。
- $m_i^{(k)}$ 是節點 i 在第 k 層的聚合信息。

常見的聚合方法包括：

- 加和 (Sum) : $m_i^{(k)} = \sum_{j \in \mathcal{N}(i)} h_j^{(k-1)}$
- 平均 (Mean) : $m_i^{(k)} = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} h_j^{(k-1)}$
- 最大 (Max) : $m_i^{(k)} = \max_{j \in \mathcal{N}(i)} h_j^{(k-1)}$

b. 特徵更新

聚合信息後，節點會更新自己的特徵。更新規則可以表示為：

$$h_i^{(k)} = \sigma \left(W^{(k)} \cdot m_i^{(k)} + b^{(k)} \right)$$

其中：

- $h_i^{(k)}$ 是節點 i 在第 k 層的更新後的特徵。
- $W^{(k)}$ 是第 k 層的權重矩陣， $b^{(k)}$ 是偏置項。
- σ 是非線性激活函數，通常使用 ReLU。

3. 圖卷積網絡 (GCN)

圖卷積網絡 (GCN) 是 GNN 中最常見的一種變體。在 GCN 中，聚合操作通常通過歸一化的鄰接矩陣來進行，特徵更新則通過一層線性變換來實現。

a. GCN 層的傳播規則

GCN 層的傳播規則可以表示為：

$$H^{(k)} = \sigma \left(\hat{A} H^{(k-1)} W^{(k)} \right)$$

其中：

- $\hat{A} = D^{-1/2} A D^{-1/2}$ 是歸一化的鄰接矩陣， D 是度數矩陣（對角矩陣， $D_{ii} = \sum_j A_{ij}$ ）。
- $W^{(k)}$ 是第 k 層的權重矩陣。
- σ 是非線性激活函數，通常使用 ReLU。

4. 圖注意力網絡 (GAT)

在 圖注意力網絡 (GAT) 中，聚合過程引入了注意力機制，使得每個節點能根據其鄰居的特徵自動分配不同的權重。更新規則可以表示為：

$$h_i^{(k)} = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W^{(k)} h_j^{(k-1)} \right)$$

其中：

- α_{ij} 是節點 i 和節點 j 之間的注意力係數，這些係數通常通過自注意力機制來計算。

注意力係數 α_{ij} 的計算方式為：

$$\alpha_{ij} = \frac{\exp \left(\text{LeakyReLU} \left(a^T [W h_i^{(k-1)} || W h_j^{(k-1)}] \right) \right)}{\sum_{k \in \mathcal{N}(i)} \exp \left(\text{LeakyReLU} \left(a^T [W h_i^{(k-1)} || W h_k^{(k-1)}] \right) \right)}$$

其中 a 是可學習的注意力向量， $||$ 表示拼接操作。

5. 最終輸出

經過多層的聚合和更新，最終的節點表示可以用於各種任務：

- **節點分類**：將最終的節點表示 $H^{(K)}$ 輸入到 softmax 層進行節點分類。
- **圖級任務**：可以對所有節點的特徵進行池化，將節點特徵聚合為單一的圖表示，用於圖分類等任務。

GNN 流程總結

1. 輸入：圖 $G = (V, E)$ 、節點特徵 X 、鄰接矩陣 A 。
2. 信息傳遞：從鄰居聚合特徵。
3. 更新：根據聚合結果更新節點特徵。
4. 輸出：最終的節點特徵或圖級特徵，用於下游任務。

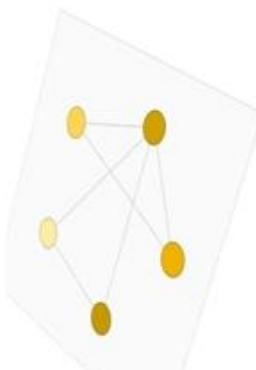
Graph Neural Network



A Gentle Introduction to Graph Neural Networks

Neural networks have been adapted to leverage the structure and properties of graphs. We explore the components needed for building a graph neural network - and motivate the design choices behind them.

Layer 0



Layer 1



Layer 2

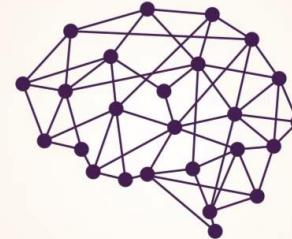


Layer 3



Graph Neural Network

15:50



GRAPH NEURAL NETWORKS

GraphGAN 將 GAN 的生成器 (Generator) 和判別器 (Discriminator) 結合到圖結構數據上：

1. 生成器 (Generator) :

- 用於生成節點之間的潛在關係，例如推測未知邊的存在與否。
- 通常通過隨機抽樣生成接近真實分布的節點或邊。

2. 判別器 (Discriminator) :

- 判斷生成的節點關係是否為真實樣本。
- 通過學習分辨生成的與真實的圖結構特徵，提升生成器的性能。

GraphVAE 的架構可以分為兩個主要部分：編碼器（Encoder）和解碼器（Decoder）。

1. 編碼器（Encoder）

將輸入圖轉換為潛在空間的分佈表示。

- 使用圖神經網路（GNN，如GCN或GIN）來學習節點的嵌入，捕捉圖結構和節點屬性。
- 將嵌入進一步轉換為潛在空間的均值 μ 和方差 σ^2 ，分別定義潛在變量 z 的分佈：

$$q(z|G) = \mathcal{N}(z; \mu, \text{diag}(\sigma^2)),$$

其中 G 是輸入圖， z 是潛在表示。

2. 解碼器（Decoder）

從潛在空間的樣本 z 重建圖的結構和特徵。

- 解碼器通常建模圖的鄰接矩陣 A 和節點特徵矩陣 X 的生成過程：

$$p(A, X|z) = p(A|z) \cdot p(X|A, z).$$

- $p(A|z)$ 通常使用多層感知機（MLP）或生成模型來生成鄰接矩陣。

Graph Diffusion Models 的核心思想、框架與應用介紹：

1. 正向擴散過程 (Forward Diffusion Process) :

- 將圖數據 $G = (A, X)$ 的結構 (鄰接矩陣 A) 和節點特徵 X 添加隨機噪聲，逐步轉換為標準高斯分佈。
- 每一步的轉換遵循以下遞推公式：

$$q(G_t | G_{t-1}) = \mathcal{N}(G_t; \sqrt{1 - \beta_t} G_{t-1}, \beta_t I),$$

其中 G_t 表示第 t 步的圖狀態， β_t 是每一步的擴散係數。

2. 反向生成過程 (Reverse Diffusion Process) :

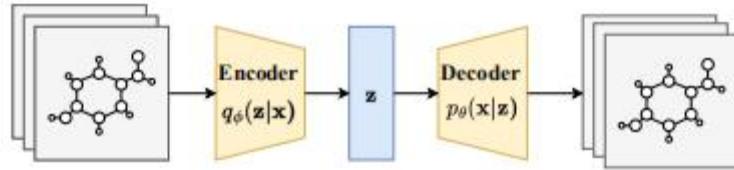
- 從噪聲圖 G_T (接近高斯分佈) 逐步去噪，還原到原始圖 G_0 。
- 每一步生成依賴於學習到的去噪分佈：

$$p_\theta(G_{t-1} | G_t) = \mathcal{N}(G_{t-1}; \mu_\theta(G_t, t), \Sigma_\theta(t)),$$

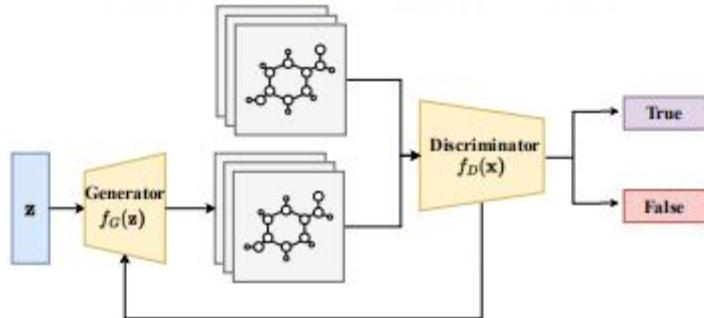
其中 μ_θ 和 Σ_θ 是參數化模型 (如 GNN) 學習到的去噪均值和方差。

3. 圖結構與特徵的建模：

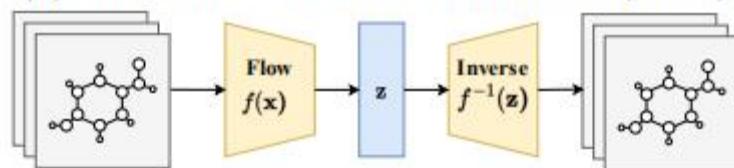
- A 和 X 可分別進行擴散建模，也可以作為聯合分佈來建模。



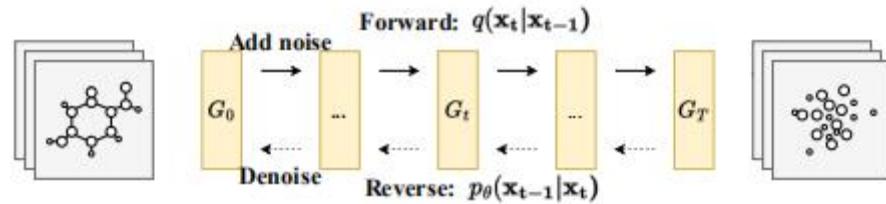
(a) Variational AutoEncoders



(b) Generative Adversarial Networks (GAN)



(c) Normalizing Flows



(d) Diffusion models

Figure 1: Deep Generative Models on Graphs.

圖變換器 (Graph Transformer)

圖變換器是一種將圖神經網路 (Graph Neural Networks, GNNs) 與變換器 (Transformers) 結合的模型架構，專為處理圖結構數據而設計。它利用變換器的自注意力機制來學習節點與節點之間的長距離依賴關係，同時保持圖結構的特性。

核心技術概念

1. 圖數據的結構表示

- 紿定一個圖 $G = (V, E)$ ，其中：
 - V 是節點集合。
 - E 是邊集合。
 - 每個節點 v_i 和邊 e_{ij} 可附帶特徵向量 x_i 和 e_{ij} 。
- Graph Transformer 的目標是學習節點或圖的嵌入表徵，用於下游任務。

2. 自注意力機制 (Self-Attention)

- Graph Transformer 利用自注意力來學習節點之間的關係：

$$\alpha_{ij} = \text{Softmax} \left(\frac{(h_i W_Q)(h_j W_K)^T}{\sqrt{d_k}} \right)$$

- h_i : 節點 i 的特徵向量。
- W_Q, W_K : 查詢和鍵的線性投影矩陣。
- d_k : 特徵維度的縮放因子。

3. 位置編碼 (Positional Encoding)

- 為了保留圖結構資訊，Graph Transformer 在節點特徵中加入位置編碼，常基於：
 - 節點之間的圖距離。
 - 拉普拉斯矩陣的特徵值和特徵向量。

4. 全局與局部信息的結合

- 自注意力能捕捉全局關係，但為了增強局部鄰域的特性，模型會將 GNN 的消息傳遞機制與自注意力結合。

模型架構

1. 輸入層

- 節點特徵和邊特徵嵌入：

$$h_i^{(0)} = f(x_i), \quad e_{ij}^{(0)} = g(e_{ij})$$

2. 多頭自注意力層

- 通過多頭自注意力學習節點的多樣性關係：

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W_O$$

3. 鄰域聚合與更新

- 將鄰居的信息聚合到節點：

$$h_i^{(l+1)} = \text{LayerNorm} \left(h_i^{(l)} + \text{Attention}(h_i^{(l)}) \right)$$

4. 讀出層

- 將所有節點嵌入匯總，生成圖級或節點級表徵，用於分類或迴歸等任務。

比較項目	GNN (圖神經網路)	Graph Transformer
核心機制	基於「鄰域聚合」(Message Passing)。	基於「自注意力機制」(Self-Attention)。
資訊傳遞方式	依賴節點及其鄰居的局部結構，進行逐層聚合更新。	節點之間透過全局注意力機制直接進行資訊傳遞。
依賴範圍	主要依賴局部鄰域資訊，難以捕捉長距離依賴性。	能捕捉節點間的長距離依賴性，全局資訊表現更佳。
計算成本	計算效率較高，適用於大規模圖數據。	計算成本較高，注意力機制隨節點數量成平方增長。
圖結構保留	天然適用於圖結構數據，局部鄰域資訊保留效果佳。	需要加入位置編碼 (Positional Encoding) 來保留圖結構。
訓練與過平滑	在深層網路中容易出現「過平滑」(Over-Smoothing) 問題。	自注意力機制可緩解過平滑，但易受計算資源限制。
特徵表現能力	鄰域聚合方式主要關注鄰居資訊，特徵提取能力較局限。	自注意力使節點能與所有其他節點交互，學習更豐富的特徵。
應用場景	適用於圖分類、節點分類、鏈結預測等局部結構任務。	適用於分子屬性預測、社群分析等全局依賴性較強的任務。

Graph Transformer

