GNN-Transformer：圖神經網路如何增強表格資料的預測能力

指導教授：李政德
專題成員：侯丞祐
開發工具：PyTorch、PyG、PyTorch Frame、Cython、NumPy
測試環境：Ubuntu 24.04 LTS、Python 3.11.9

一、簡介：

隨著近幾年深度學習模型在表格數據上被陸續提出，ExcelFormer 雖是目前最新且表現卓越的 Transformer-based 模型，但其僅考慮資料內部的特徵交互而容易忽略數據上的整體結構分佈。因此，本專題致力於結合擅長捕捉全局關係的圖神經網路（GNN）以增強 ExcelFormer 在表格數據上的預測能力，並以此分成三大面向討論：

1. **表格資料的建圖策略**

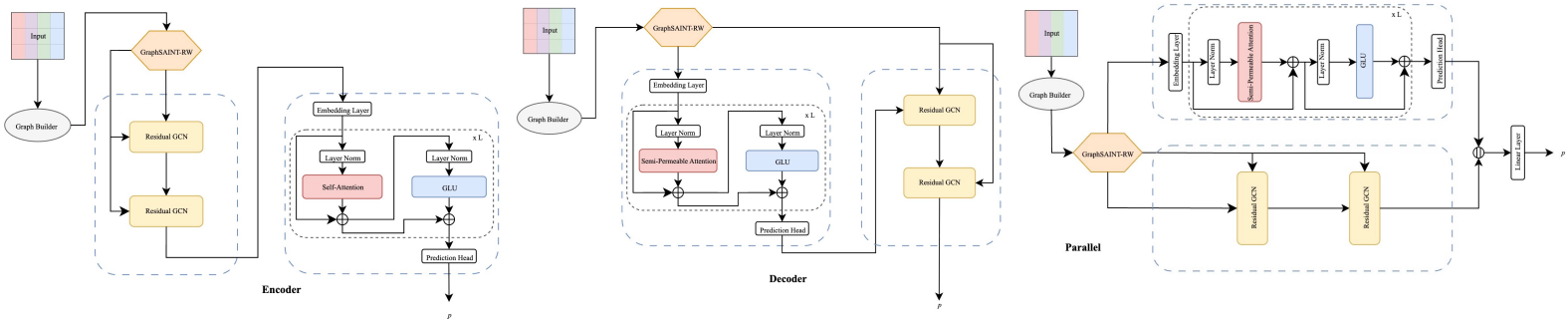
我們將在專題上呈現 Multi-thread Adaptive Mutual kNN 建圖法，藉由動態調整 k 值與 Mutual 性質構圖、隨機採樣決定距離閥值以及用多執行緒計算節點間距離和構建邊（後者透過 Cython 實作跳脫 GIL 限制）。

2. **適合的 Mini-batch 訓練方式**

在訓練上本專題藉由 GraphSAINT-RW 進行子圖採樣訓練，在記憶體開銷減少的同時，訓練速度能比原始 ExcelFormer 還快，甚至拉高整體預測能力上限。

3. **在表格數據上，GNN 適合的角色定位**

下圖皆為系統架構圖：



- Encoder：GNN 對資料生成具有全局感知的嵌入後，送入 ExcelFormer 完成後續下游任務。
- Decoder：ExcelFormer 對各筆資料產生特徵表示後，將其視為節點，再讓 GNN 進行訊息傳播和預測。
- Parallel：GNN 和 ExcelFormer 分別平行學習全局結構關係和內部特徵交互，並將兩者整合以進行預測。

二、測試結果：

本專題旨在探討將 GNN 結合至 ExcelFormer 在表格數據上的提升性，在實驗設計上涵蓋二元分類任務和回歸任務作為評估，並且在模型訓練上會固定 PyTorch Frame 在 ExcelFormer 的預設超參數配置，僅針對以下超參數進行調整：

- Hidden Channel of Residual GCN：設為{64, 128, 256}，作為 GNN 隱藏層維度大小，將影響模型的學習能力。
- Number of GraphSAINT-RW Steps per Epoch：設為{10, 15}，決定每個訓練週期會抓多少個子圖進行訓練。
- Batch Size of GraphSAINT-RW per Step：依照資料總筆數進行相對應調整，此設定會配合固定步長為2的隨機遊走策略，用以控制每個子圖的規模。

	Binary Classfication (AUC)					Regression (RMSE)				
	AdultCensusIncome	Mushroom	BankMarketing	MagicTelescope	bank-marketing	Bike_Sharing_Demand	Brazilian_houses	cpu_act	elevators	house_sales
ExcelFormer	0.871 ± 0.023	0.978 ± 0.043	0.887 ± 0.025	0.887 ± 0.047	0.862 ± 0.006	0.263 ± 0.008	0.015 ± 0.003	0.141 ± 0.000	0.278 ± 0.002	0.306 ± 0.001
Encoder	0.899 ± 0.004	**1.000 ± 0.000**	0.892 ± 0.005	0.931 ± 0.009	0.863 ± 0.005	0.313 ± 0.005	0.117 ± 0.005	0.144 ± 0.001	0.288 ± 0.001	0.345 ± 0.003
Decoder	**0.919 ± 0.001**	**1.000 ± 0.000**	**0.938 ± 0.002**	**0.951 ± 0.001**	**0.887 ± 0.004**	0.275 ± 0.007	0.019 ± 0.003	0.136 ± 0.001	0.282 ± 0.002	0.306 ± 0.001
Parallel	0.917 ± 0.006	**1.000 ± 0.000**	0.934 ± 0.001	0.945 ± 0.004	0.885 ± 0.003	**0.262 ± 0.004**	**0.014 ± 0.000**	**0.131 ± 0.001**	**0.277 ± 0.001**	**0.305 ± 0.001**

表 1 ：結合 GNN 架構對比原始 ExcelFormer 的學習結果

1. GNN 扮演角色在表格資料學習上的影響

根據表1，GNN 作為 Encoder 雖能產生全域感知的嵌入，卻會被 ExcelFormer 的特徵交互弱化，效果不如預期。GNN 擔任 Decoder 時，ExcelFormer 先提取特徵後進行有效的節點間訊息傳播，在二元分類任務的決策邊界上佔有優勢。

在 Parallel 架構中，GNN 和 ExcelFormer 獨立學習，融合後能兼顧全局與局部關係、提升精度，在回歸任務上表現最好，整體表現優於原始 ExcelFormer。

2. Mini-batch 優於 Full-batch 訓練、最終表現

由於 GraphSAINT-RW 是子圖採樣訓練，根據圖1顯示，Mini-batch 僅需較少的訓練週期就能穩定收斂，並在記憶體開銷上更有效率。根據表2的實驗，即使 GNN 隱藏層維度皆為64，其預測能力仍比 Full-batch 更好，更不容易發生 OOM。從表3來看，子圖採樣訓練對於較高總筆數的資料也有訓練速度快、預測表現好的優勢。

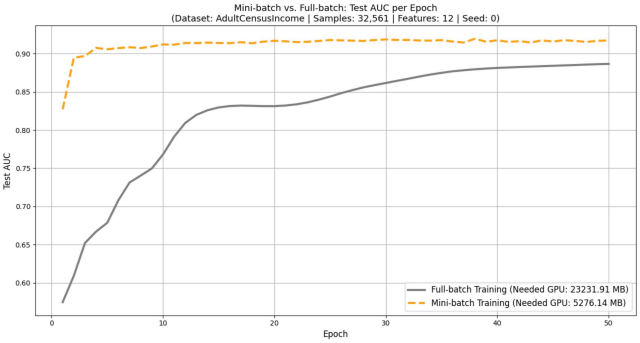


圖1：批次訓練的過程差異

	AdultCensusIncome	Mushroom	BankMarketing	MagicTelescope	bank-marketing
Mini-batch*	**0.919 ± 0.001**	**1.000 ± 0.000**	**0.938 ± 0.002**	**0.950 ± 0.001**	**0.887 ± 0.004**
Full-batch*	0.888 ± 0.004	0.999 ± 0.001	OOM	0.906 ± 0.009	0.851 ± 0.005

表2：批次訓練的最終表現差異

3. 優化構圖策略執行效率

按照圖2，在距離閥值使用隨機採樣降低執行時間以及最終將資料格式轉成 Tensor 外，距離矩陣的計算和邊的構建都可用多執行緒加速，並對後續訓練影響甚小。

	Diabetes130US (71090 rows × 7 columns)	
	AUC	Training Time
ExcelFormer	0.616 ± 0.023	269.3 ± 9.4 s
DECODER	**0.649 ± 0.001**	97.5 ± 2.2 s

表3：使用 GraphSAINT-RW 在結合 GNN 架構加強預測能力上限和節省訓練時間

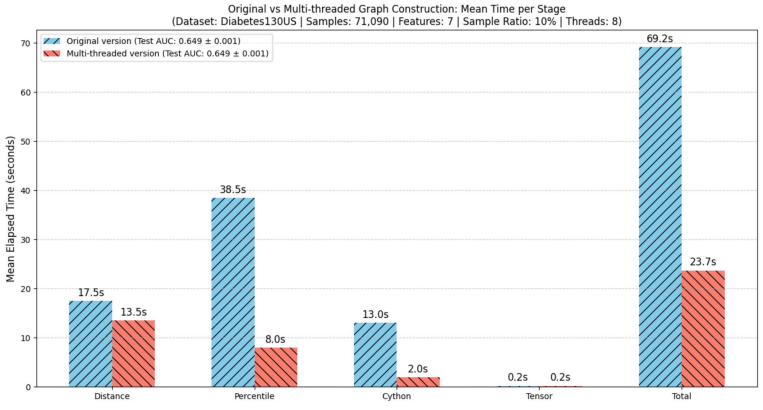


圖2：原始演算法和多執行緒版本演算法，在不同階段（包括總體）的執行時間