Comparative Analysis of Spatial and Spectral Methods in GNN for Power Flow in Electrical Power Systems

Paulo A. Espinoza, Gonzalo A. Ruz, PhD.

Faculty of Engineering and Sciences, Universidad Adolfo Ibáñez, Santiago, Chile

27th Iberoamerican Congress on Pattern Recognition, Talca, Chile
November 29, 2024

Outline

Current Section

Purpose:

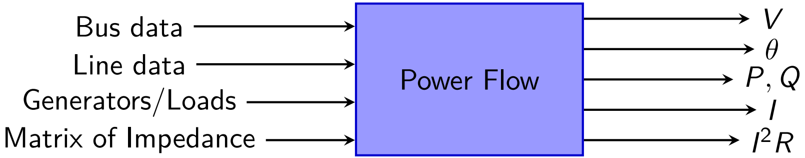
- Determine steady-state voltages, power flows, and losses in electrical systems.

Key Components:

- Buses: nodes, loads, generators, transmission lines:

Applications

- Voltage stability, contingency analysis, support operational planning, and economic dispatch.



The power flow problem is defined by a set of nonlinear equations derived from Kirchhoff's laws

$$P_i = V_i \sum_{j=1}^{N} V_j (G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij}), \tag{1}$$

$$Q_i = V_i \sum_{j=1}^{N} V_j (G_{ij} \sin\theta_{ij} - B_{ij} \cos\theta_{ij}), \tag{2}$$

$$\Delta P_i = P_i^{\text{specified}} - \boxed{V_i} \sum_{j=1}^{N} \boxed{V_j} (G_{ij} \cos \boxed{\theta_{ij}} + B_{ij} \sin \boxed{\theta_{ij}}) \tag{3}$$

$$\Delta Q_i = Q_i^{\text{specified}} - \boxed{V_i} \sum_{j=1}^{N} \boxed{V_j} (G_{ij} \sin \boxed{\theta_{ij}} - B_{ij} \cos \boxed{\theta_{ij}}) \tag{4}$$

Limitations of Traditional Methods (e.g., Newton-Raphson):

- High computational cost due to solving large nonlinear equation systems. Inefficient for large-scale networks, especially under real-time constraints.

Approximate Methods (e.g., DC Flow)

- Fast computational times. Lower accuracy is due to neglect of reactive power and assuming small voltage angle differences.

GNN: Combines the structural properties of power grids with machine learning, including physical constraints, offering efficient modeling and acceptable accuracy in suitable time frames. [4, 5, 6, 8]

Current Section

Background

Graph Neural Networks

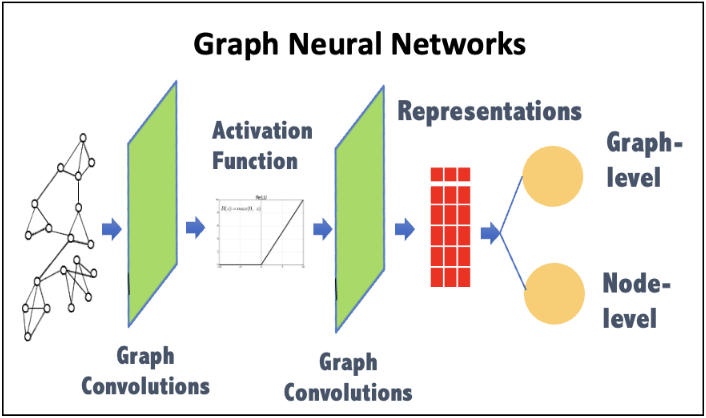


Figure: Source: https://web.stanford.edu/class/cs224w/

Background
Taxonomies of GNNs

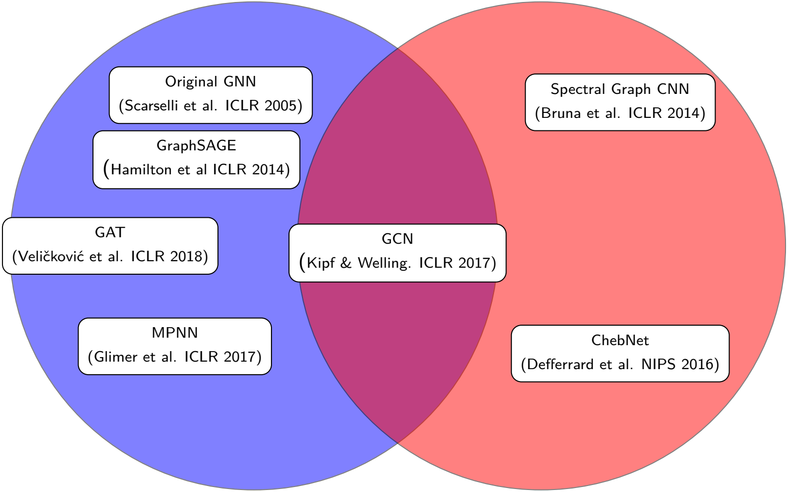


Figure: Based in https://ai.tencent.com/ailab/ml/KDD-Deep-Graph-Learning.html

In the context of GNN, a graph is represented as $G = (V, E)$, where V is a set of nodes and E is the edges connecting the nodes.

Adjacency matrix

$$A_{ij} = \begin{cases} 1 & \text{if } e_{ij} \in E \\ 0 & \text{if } e_{ij} \notin E \end{cases}$$

Degree matrix

$$D_{ij} = \begin{cases} \deg(v_i) & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Laplacian

$$L = D - A.$$

Normalized Laplacian

$$\tilde{L} = I - D^{-1/2} A D^{-1/2}.$$

The eigenvalues and eigenvectors of the $\tilde{L}$ are used to perform spectral graph convolutions. The convolution operation is expressed as:

$$g \star x = F^{-1}(F(g) \cdot F(x)) = U(U^T g \cdot U^T x), \tag{5}$$

where g is the filter in the spectral domain. The ChebNet method [1] approximates the convolution operation using Chebyshev polynomials.

$$g \star x \approx \sum_{k=0}^{K} \theta_k T_k(\tilde{L}) x, \tag{6}$$

where $\tilde{L} = \frac{2}{\lambda_{\max}} L - I$ is the rescaled Laplacian, $\lambda_{\max}$ is the largest eigenvalue of L, θ_k are the Chebyshev coefficients, and $T_k(\tilde{L})$ are defined as:

$$T_0(\tilde{L}) = I, \quad T_1(\tilde{L}) = \tilde{L}, \quad T_k(\tilde{L}) = 2\tilde{L} T_{k-1}(\tilde{L}) - T_{k-2}(\tilde{L}). \tag{7}$$

This method reduces computational complexity by avoiding the need for eigenvector computation.

Graph Convolutional Network (GCN) based in [3], simplify the expansion of Chebyshev polynomials, assuming mainly that $K = 1$ and the $\lambda_{max} = 2$. The convolution operation is now:

$$H^{(l+1)} = \sigma(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}), \tag{8}$$

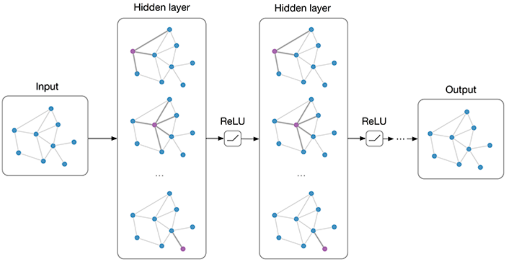


Figure: Graph Convolutional Network Multilayer

Source from: https://tkipf.github.io/graph-convolutional-networks/

Spatial Approaches operate directly on the graph structure; in this way, the convolution operation is based on the local neighborhood.

GraphSAGE based in [2], is an inductive framework that generates node embeddings by sampling and aggregating features from a node's local neighborhood. The layer-wise propagation rule is defined as:

$$h_v^{(l+1)} = \sigma\left(W^{(l)} \cdot \text{AGGREGATE}^{(l)}\left(\{h_u^{(l)}, \forall u \in \mathcal{N}(v)\}\right)\right), \qquad (9)$$

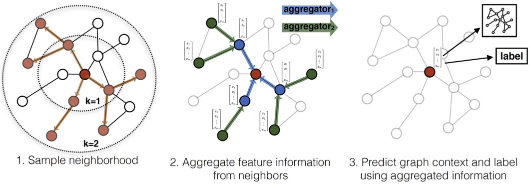
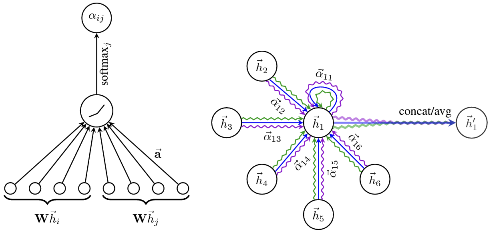


Figure: Source from: [2]

GAT [7] incorporate the attention mechanism assigning different weights to different neighbors. The attention coefficient α_{ij} is computed as:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(a^T[Wh_i \| Wh_j]\right)\right)}{\sum_{k\in\mathcal{N}(i)} \exp\left(\text{LeakyReLU}\left(a^T[Wh_i \| Wh_k]\right)\right)}, \tag{10}$$

The node representation is then updated as:



$$h_i^{(l+1)} = \sigma\Big(\sum_{j\in\mathcal{N}(i)} \alpha_{ij} W h_j\Big)$$

Current Section

Methodology

The computational frameworks used are PyTorch Geometric (PyG) and PandaPower, mainly to apply GNN algorithms and create synthetic data to support the benchmark test cases.

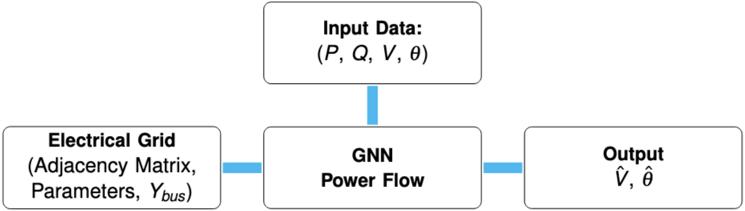


Figure: Flowchart Experiment of GNN Power Flow.

Current Section

Experiments

The test cases utilized from the pandapower library are as follows:

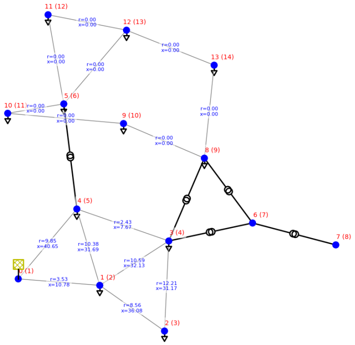


Figure: Test Case 14-Buses

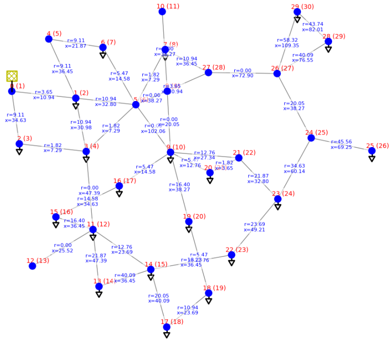


Figure: Test Case 30-Buses

In the experiments, node data is used.

Experiments

Dataset:

- Two test cases: 14 buses and 30 buses are used.
- Both datasets containing 2.000 independent observations

Objective:

Predict V and θ node level with features P, Q

Loss Function:

$$\mathcal{L}(\hat{V}, \hat{\theta}, V, \theta) = \frac{1}{N} \sum_{i=1}^{N} \left((\hat{V}_i - V_i)^2 + (\hat{\theta}_i - \theta_i)^2 \right) \tag{12}$$

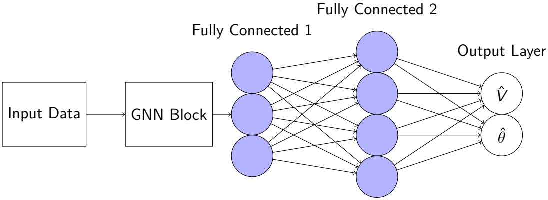


Figure: Network architecture used for all GNN blocks in the experiment.

Results

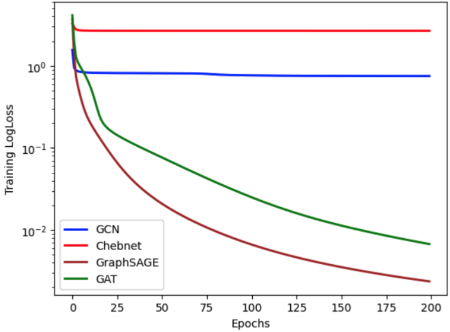


Figure: Training LogLoss Test Case 14-Bus

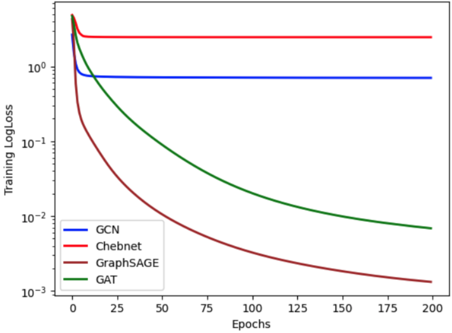


Figure: Training LogLoss Test Case 30-Bus

Results

Model	# Parameters	Taxonomy	Train Time	MAPE	R^2	RMSE	MAE
GCN	2.308	Hybrid	12 min	3,2%	0.99	0.82	0.31
ChebNet	6.120	Spectral	23 min	4,85%	0.96	1.66	0.68
GraphSAGE	2.768	Spatial	10 min	0,79%	0.99	0.11	0.05
GAT	2.768	Spatial	16 min	0,80%	0.99	0.12	0.06

Table: Performance on test case 14 - PandaPower

Model	# Parameters	Taxonomy	Train Time	MAPE	R^2	RMSE	MAE
GCN	4,900	Hybrid	13 min	3.25%	0.99	0.86	0.38
ChebNet	23,048	Spectral	27 min	4.04%	0.96	1.58	0.66
GraphSAGE	5,872	Spatial	12 min	0.53%	0.99	0.084	0.04
GAT	5,872	Spatial	17 min	0.60%	0.99	0.1	0.05

Table: Performance on test case 30 - PandaPower

Current Section

Conclusion

- **Observed Superior Performance of Spatial Methods**: Based on the results, GraphSAGE and GAT demonstrate better performance compared to spectral methods, with lower prediction errors and reduced computational demands
- **Effective Use of Local Information**: GraphSAGE and GAT effectively leverage local node-specific information, a crucial advantage in power systems where local interactions heavily influence problem dynamics.
- **Limitations of Spectral Methods**: ChebNet and GCN exhibit higher prediction errors and increased training times, likely due to their reliance on global graph structures, which can dilute critical local information.
- **Future Work**: Further studies on real-world, enhancing scalability, and ensuring the adaptability of GNN-based models for real-time applications.

Q&A

Current Section

References I

DEFFERRARD, M., BRESSON, X., AND VANDERGHEYNST, P.
Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering.
Publisher: arXiv Version Number: 3.

HAMILTON, W., YING, Z., AND LESKOVEC, J.
Inductive Representation Learning on Large Graphs.
In *Advances in Neural Information Processing Systems* (2017), I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc.

KIPF, T. N., AND WELLING, M.
Semi-Supervised Classification with Graph Convolutional Networks, Feb. 2017.
arXiv:1609.02907 [cs, stat].

OWERKO, D., GAMA, F., AND RIBEIRO, A.
Optimal Power Flow Using Graph Neural Networks.
In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Barcelona, Spain, May 2020), IEEE, pp. 5930–5934.

STOCK, S., BABAZADEH, D., AND BECKER, C.
Applications of Artificial Intelligence in Distribution Power System Operation.
IEEE Access 9 (2021), 150098–150119.

References II

Tuo, M., Li, X., and Zhao, T.
Graph Neural Network-based Power Flow Model.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y.
Graph Attention Networks.
Publisher: arXiv Version Number: 3.

Yaniv, A., Kumar, P., and Beck, Y.
Towards adoption of GNNs for power flow applications in distribution systems.
Electric Power Systems Research 216 (Mar. 2023), 109005.

Thank you!!