
Principal Neighbourhood Aggregation for Graph nets

School of Industrial and Management Engineering, Korea University

Insung Baek

Contents

- ❖ Research Purpose
- ❖ Proposed Method
- ❖ Experiments
- ❖ Conclusion

Research Purpose

PNA for Graph Nets

❖ Principal Neighbourhood Aggregation for Graph nets (NeurIPS 2020)

- University of Cambridge, InVivo AI, DeepMind가 연구를 진행, 2022년 4월 11일 기준 135회 인용됨
- 그래프 구조에서 이웃 노드로부터 충분한 정보를 추출하기 위해 여러 aggregator를 동시에 사용하고 이웃 노드의 상대적 영향력을 반영하기 위해 degree-based scaler를 사용함

Principal Neighbourhood Aggregation for Graph Nets

Gabriele Corso*
University of Cambridge
gc579@cam.ac.uk

Luca Cavalleri*
University of Cambridge
lc737@cam.ac.uk

Dominique Beaini
InVivo AI
dominique@invivoai.com

Pietro Liò
University of Cambridge
pietro.liò@cst.cam.ac.uk

Petar Veličković
DeepMind
petarv@google.com

Abstract

Graph Neural Networks (GNNs) have been shown to be effective models for different predictive tasks on graph-structured data. Recent work on their expressive power has focused on isomorphism tasks and countable feature spaces. We extend this theoretical framework to include continuous features—which occur regularly in real-world input domains and within the hidden layers of GNNs—and we demonstrate the requirement for multiple aggregation functions in this context.

Research Purpose

Research Purpose

- ❖ Graph Convolutional Network (GCN)은 그래프 구조 데이터에 대한 예측 task를 수행하는데 효과적인 모델로 알려져 있음
- ❖ 기존 연구들은 단일 task 예측 및 셀 수 있는 (countable) feature space에 초점을 두고 있음
- ❖ 본 연구에서는 실제 현실 세계에서 발생할 수 있는 그래프 문제를 풀기 위해 multi-task 및 연속적인 (continuous) feature space로 기존 연구의 프레임워크를 확장하고자 함
- ❖ 본 연구에서 제안한 Principal Neighbourhood Aggregation (PNA)에서는 1) multiple aggregators 와 2) degree-scaler를 사용해 위와 같은 프레임워크 확장이 가능하다고 설명함

Proposed Method

Proposed aggregators

- ❖ 기존 연구에서는 mean, sum, max와 같은 단일 aggregator만 사용하였음
- ❖ 하지만 단일 aggregator로는 주변 node의 다른 특징을 구별하지 못하는 한계점이 존재
- ❖ Figure 1에서 첫번째 예시에서 주변 node의 평균은 '2'로 graph 1과 graph 2 모두 같지만, 실제로는 각 node의 값이 다르기 때문에 각 node의 특징을 정확히 요약하지 못했음을 의미



주황색으로 표시된 부분이 aggregation이 제대로 안 된 경우를 의미

Figure 1: Examples where, for a single GNN layer and continuous input feature spaces, some aggregators fail to differentiate between neighbourhood messages.

Proposed Method

Proposed aggregators

- ❖ 따라서 본 연구에서는 주변 node의 정확한 정보를 요약하기 위해 multiple aggregators를 사용해야 함을 제안하고 있음
- ❖ 특히 continuous input/feature spaces에서는 작은 변화도 aggregators에 중요한 영향을 주기 때문에 multiple aggregators를 사용하는 것이 더 중요함

(1) Mean aggregation

$$\mu(X) = \mathbb{E}[X] \quad , \quad \mu_i(X^l) = \frac{1}{d_i} \sum_{j \in N(i)} X_j^l$$

(2) Maximum and minimum aggregations

$$\max_i(X^l) = \max_{j \in N(i)} X_j^l \quad , \quad \min_i(X^l) = \min_{j \in N(i)} X_j^l$$

(3) Standard deviation aggregation

$$\sigma(X) = \sqrt{\mathbb{E}[X^2] - \mathbb{E}[X]^2} \quad , \quad \sigma_i(X^l) = \sqrt{\text{ReLU} \left(\mu_i(X^{l^2}) - \mu_i(X^l)^2 \right) + \epsilon}$$

(4) Normalized moments aggregation

$$M_n(X) = \sqrt[n]{\mathbb{E}[(X - \mu)^n]} \quad , \quad n > 1$$



위 3개 aggregation으로 충분하지 않을 때 사용

Proposed Method

Degree-based scalers

- ❖ 본 연구에서는 주변 이웃 node의 영향력을 더 정확하게 설명하기 위해 로그 스케일러 (logarithmic scalers)를 사용함
- ❖ 예를 들어, 이웃 node인 A, B, C 값이 500만, 100만, 100인 경우 실제 값은 B와 C가 A와 B보다 가깝다고 평가될 것이지만 실제 A, B, C가 지니는 상대적인 영향력을 정확하게 판단하기 어렵다는 한계가 존재함

We propose the logarithmic scaler S_{amp} presented in Equation 5, where δ is a normalization parameter computed over the training set, and d is the degree of the node receiving the message.

$$S_{\text{amp}}(d) = \frac{\log(d+1)}{\delta} \quad , \quad \delta = \frac{1}{|\text{train}|} \sum_{i \in \text{train}} \log(d_i + 1) \quad (5)$$

We further generalize this scaler in Equation 6, where α is a variable parameter that is negative for attenuation, positive for amplification or zero for no scaling. Other definitions of $S(d)$ can be used—such as a linear scaling—as long as the function is injective for $d > 0$.

$$S(d, \alpha) = \left(\frac{\log(d+1)}{\delta} \right)^\alpha \quad , \quad d > 0, \quad -1 \leq \alpha \leq 1 \quad (6)$$

Proposed Method

Combined aggregation

❖ Figure 2에서는 본 연구에서 제안하는 PNA 방식을 보여줌

- 1) Graph data에서 주변 node에 대해 multiple aggregators를 적용해 요약 값 산출
- 2) 각 요약 값에 대해 여러 scalers를 적용해 값 도출해 모델의 입력 데이터로 사용

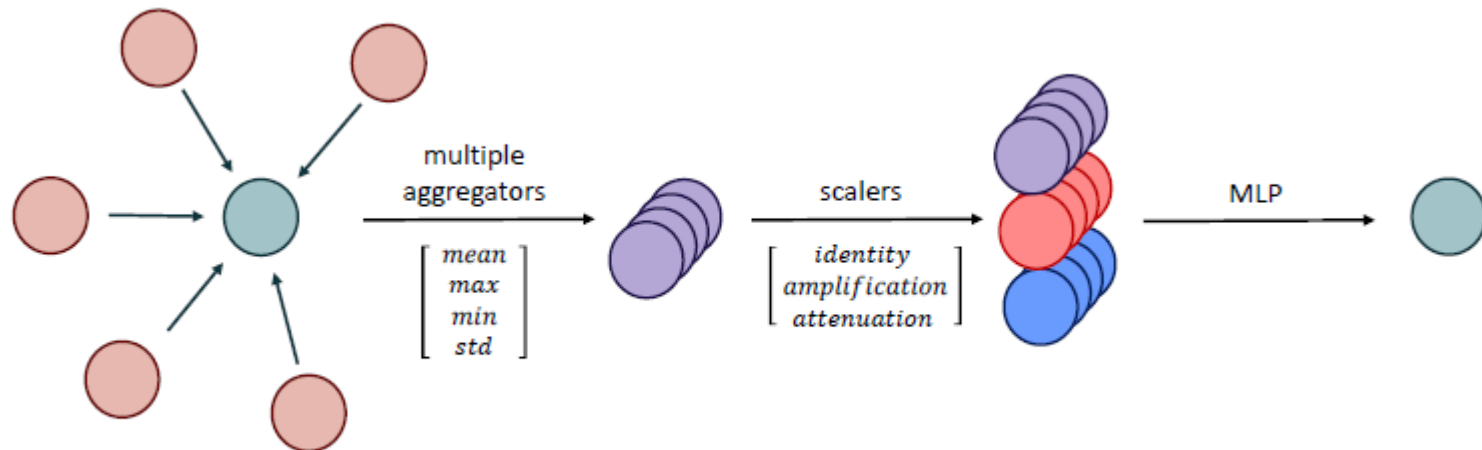


Figure 2: Diagram for the Principal Neighbourhood Aggregation or PNA.

Proposed Method

Architecture

- ❖ Figure 3은 본 연구에서 제안하는 PNA 방식을 적용해 node labels과 graph labels를 동시에 예측하는 multi-task 문제를 해결하기 위한 전반적인 모델 구조를 보여줌

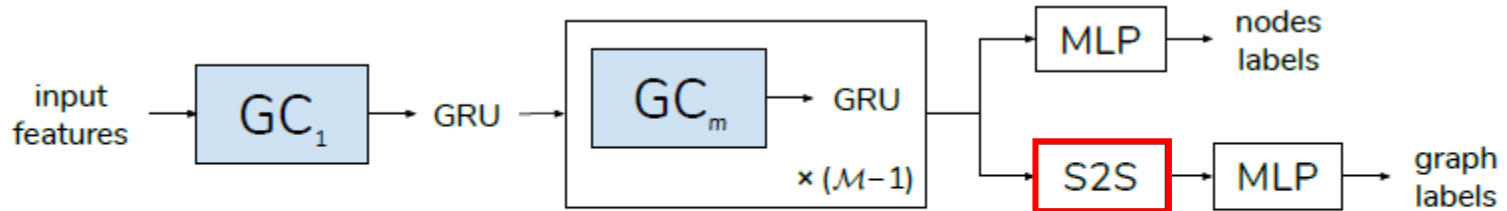


Figure 3: Layout of the architecture used. When comparing different models, the difference lies only in the type of graph convolution used in place of GC_1 and GC_m .

```
CLASS Set2Set ( in_channels: int, processing_steps: int, num_layers: int = 1 ) \[source\]
```

The global pooling operator based on iterative content-based attention from the “Order Matters: Sequence to sequence for sets” paper

$$\begin{aligned} \mathbf{q}_t &= \text{LSTM}(\mathbf{q}_{t-1}^*) \\ \alpha_{i,t} &= \text{softmax}(\mathbf{x}_i \cdot \mathbf{q}_t) \\ \mathbf{r}_t &= \sum_{i=1}^N \alpha_{i,t} \mathbf{x}_i \\ \mathbf{q}_t^* &= \mathbf{q}_t \parallel \mathbf{r}_t, \end{aligned}$$

Results

Multi-task benchmark

- ❖ Node labels과 graph labels이 각 3개씩 존재하는 Multi-task benchmark data 생성해 평가 진행
 - Multi-task benchmark data 생성에 대한 자세한 내용은 논문 내 Appendix G 참조
- ❖ Scalers를 적용한 PNA 방법이 모든 labels에 대해서 \log MSE 값이 가장 낮았음을 확인
- ❖ 단일 aggregation을 사용한 MPNN 대비해서도 PNA 방법이 가장 성능이 좋았음을 확인

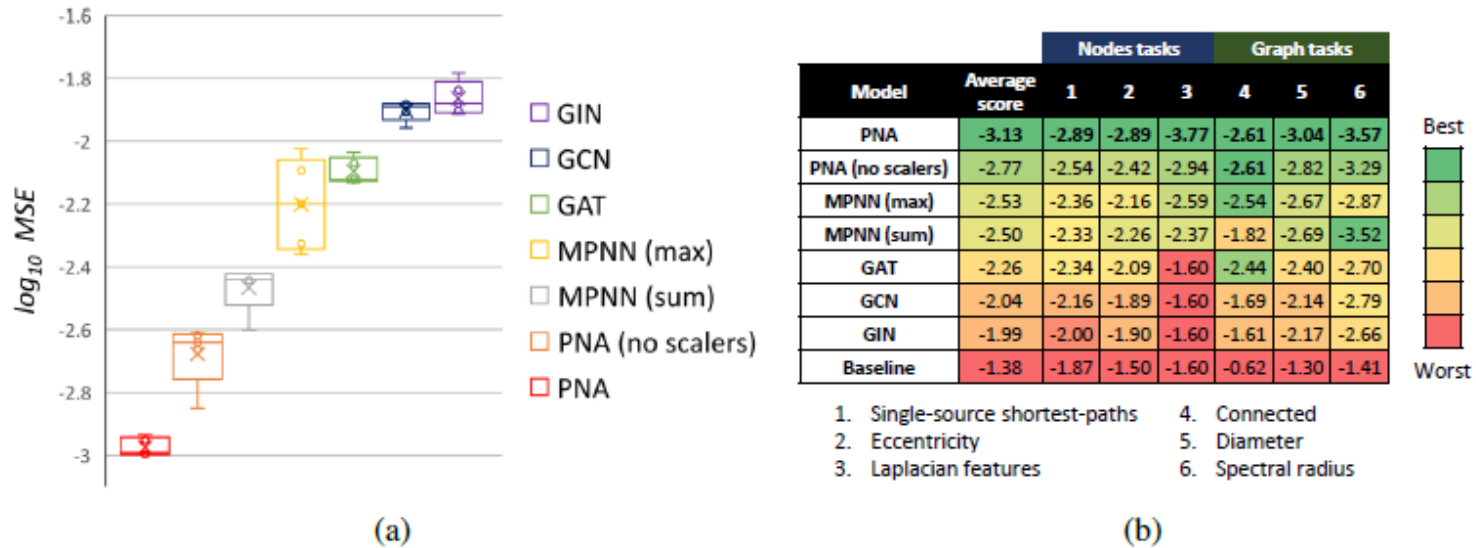


Figure 4: Multi-task benchmarks for different GNN models using the same architecture and various near-optimal hyper-parameters. (a) Distribution of the \log_{10} MSE errors for the top 5 performances of each model. (b) Mean \log_{10} MSE error for each task and their combined average.

Results

Real-world benchmark

- ❖ 화학 분야 데이터 (ZINC, MolHIV)와 컴퓨터 비전 데이터 (CIFAR10, MNIST)으로 실험 진행
- ❖ 전반적으로 scalers를 적용한 PNA 방법의 성능이 가장 좋았지만 특히 실제 그래프 구조에 더 적합한 화학 분야 데이터에서 더 높은 성능 향상을 확인할 수 있었음
- ❖ 제안한 PNA 방법이 그래프 구조 데이터에서 더 좋은 특징을 추출하고 있음을 확인

		ZINC		CIFAR10		MNIST		MolHIV
Model		No edge features MAE	Edge features MAE	No edge features Acc	Edge features Acc	No edge features Acc	Edge features Acc	No edge features % ROC-AUC
Dwivedi <i>et al.</i> and Xu <i>et al.</i> papers	MLP	0.710±0.001		56.01±0.90		94.46±0.28		
	GCN	0.469±0.002		54.46±0.10		89.99±0.15		76.06±0.97
	GIN	0.408±0.008		53.28±3.70		93.96±1.30		75.58±1.40
	DiffPool	0.466±0.006		57.99±0.45		95.02±0.42		
	GAT	0.463±0.002		65.48±0.33		95.62±0.13		
	MoNet	0.407±0.007		53.42±0.43		90.36±0.47		
	GatedGCN	0.422±0.006	0.363±0.009	69.19±0.28	69.37±0.48	97.37±0.06	97.47±0.13	
Our experi- ments	MPNN (sum)	0.381±0.005	0.288±0.002*	65.39±0.47	65.61±0.30	96.72±0.17	96.90±0.15	
	MPNN (max)	0.468±0.002	0.328±0.008*	69.70±0.55	70.86±0.27	97.37±0.11	97.82±0.08	
	PNA (no scalers)	0.413±0.006	0.247±0.036*	70.46±0.44	70.47±0.72	97.41±0.16	97.94±0.12	78.76±1.04
	PNA	0.320±0.032	0.188±0.004*	70.21±0.15	70.35±0.63	97.19±0.08	97.69±0.22	79.05±1.32

Figure 6: Results of the PNA and MPNN models in comparison with those analysed by Dwivedi *et al.* and Xu *et al.* (GCN[15], GIN[6], DiffPool[32], GAT[16], MoNet[33] and GatedGCN[34]). * indicates the training was conducted with additional patience to ensure convergence.

Conclusion

❖ Conclusion

- 기존 GCN은 그래프 구조의 데이터를 분석하는데 효과적이었지만 그래프 구조의 기본적인 속성을 정확하게 캡처해 활용하려는 방안에 대한 연구는 적었음
- Aggregation 방법으로 그래프 구조에서 주변 node의 좋은 특징을 추출하려는 시도는 있었지만 단일 aggregation 방법으로는 주변 node의 특징을 정확하게 추출하는 데는 한계가 존재함
- 본 연구에서는 mean, max, min, std 등 다양한 aggregation을 함께 활용해 주변 node의 좋은 특징을 더 정확하게 추출함
- 또한 주변 node의 상대적인 영향력을 더 정확하게 반영하기 위해 Degree-based scalers를 사용함
- 최종적으로 multiple aggregators와 degree-based scaler를 구성된 PNA 방식을 제안함
- Multi-task benchmark와 real-world benchmark에 대한 실험 결과를 통해 PNA 방식이 그래프 구조의 데이터 특징을 잘 추출해 모델의 성능 향상에 기여한다는 것을 확인함

Thank You

Appendix

Appendix

Reference

- ❖ Corso, G., Cavalleri, L., Beaini, D., Liò, P., & Veličković, P. (2020). Principal neighbourhood aggregation for graph nets. *Advances in Neural Information Processing Systems*, 33, 13260-13271.