

Nonogram-Solver: 그래프 신경망 접근 방식

*진하은, **신민철, ***최태영, 김무철*

중앙대학교 컴퓨터공학과

*he9273@cau.ac.kr, **mcshin1648@gmail.com, ***tychoe9601@gmail.com, kimm@cau.ac.kr*

Nonogram-Solver: Graph Neural Network Approach

*Chun Ha Eun, **Shin Min Cheol, ***Choe Tae Young, Kim Mu Cheol*

Chung-Ang University

요약

노노그램 퍼즐은 다양한 관객에게 인기 있는 오락일 뿐만 아니라 논리적 추론과 인지 능력을 향상시킵니다. 특정 노드(픽셀)의 값이나 상태는 고유한 특성에 의해 결정될 뿐만 아니라 주변 노드의 구조에 의해 크게 영향을 받습니다. 이러한 특성은 그래프 신경망(GNN)의 본질적인 특성과 일치하여 노노그램 퍼즐을 해결하는 데 적합한 접근 방식이 됩니다. 이 연구에서는 노노그램 퍼즐을 노드 분류 문제로 프레임화하여 GNN을 사용해 노노그램 그래프의 구조와 특성을 활용하여 노드에 적절한 레이블을 예측하고 할당하는 접근 방식을 제안합니다. 이 탐구는 노노그램 퍼즐의 속성에 대한 이해를 제공하며, 전체 해결 경험을 향상시키는 효율적인 해결 전략을 개발하는 데 도움을 줍니다.

I. 서론

노노그램(Nonogram)은 한지(Hanjie), 숫자 채우기(Paint by Numbers), 픽크로스(Picross) 등으로 알려진 퍼즐 게임으로, 격자 기반 보드를 채우고 숨겨진 패턴을 드러낸다. 이 퍼즐은 픽셀과 이들의 연결 관계(엣지)로 구성된 그래프로 표현되며, 각 노드는 픽셀을, 엣지는 인접한 픽셀 간의 관계를 나타낸다. 노노그램은 논리적 사고와 문제 해결 능력이 필요한 NP-완전 문제로 알려져 있다. [1]

노노그램을 노드 분류 문제로 접근하여, 그래프의 구조와 특성을 활용해 각 노드에 적절한 레이블을 예측하고 할당하는 방식은 다양한 응용 가능성과 고급 해결 기술 개발에 기여할 수 있다.

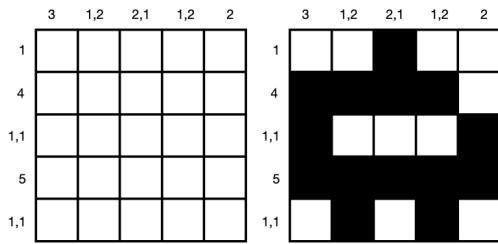


그림 1. 노노그램 예시

II. 관련연구

노노그램 퍼즐의 목적은 그리드 측면의 숫자를 기반으로 각 셀의 색상이나 음영을 결정하는 것이다. [2] 퍼즐은 직사각형 격자로 제공되며, 상단과 왼쪽에 있는 숫자 단서는 연속된 색칠된 칸 수를 나타낸다. 여러 숫자가 있는 경우, 그룹 사이에 빈 칸이 있어야 한다. 일반적으로 해결책은 픽셀화된 숨겨진 그림을 드러낸다.

노노그램은 퍼즐의 특성 때문에 다양한 알고리즘 접근 방식을 사용해 해결할 수 있다.

A. 전통적인 노노그램 해결 방법

논리적 규칙: 특정 논리 규칙과 백트래킹을 사용하여 결정되지 않은 셀을 효율적으로 해결한다. [3]

동적 프로그래밍: 동적 프로그래밍([2]) 방법을 사용해 퍼즐을 해결한다.

진화 알고리즘: 진화 알고리즘([4])을 사용해 여러 솔루션 중 하나의 솔루션을 제공한다.

선을 제공한다.

B. 그래프 신경망 (GNN)

그래프 신경망(GNN)은 그래프 구조 데이터를 이해하고 패턴을 추출하는 유망한 딥러닝 아키텍처다. GNN의 설계는 노드, 가장자리 및 전체 그래프의 정교한 표현 학습을 허용한다.

초기 2010년대에 등장한 그래프 컨볼루션 네트워크(GCN[7])는 비유리드 도메인인 그래프에 컨볼루션 연산을 적용하는 시도로 시작했다. 2016년에 Thomas Kipf와 Max Welling이 스펙트럼 그래프 컨볼루션의 1차 근사를 사용해 더 간소화된 버전을 도입했다. 그래프 어텐션 네트워크(GAT[6])는 어텐션 메커니즘을 포함해 인접 노드의 중요성을 동적으로 평가한다. GraphSAGE[7]는 인접 노드의 특징을 결합하는 새로운 방법을 제공하여 풍부한 노드 표현을 만든다. GraphSAGE는 전체 그래프나 노드의 하위 집합에서 학습을 허용해 적응성을 향상시킨다.

C. GNN을 사용한 NP-난해 퍼즐 해결

GNN을 사용해 Sudoku, Kakuro, Nonogram과 같은 NP-난해 문제를 해결하는 것은 최근의 흥미로운 연구 방향이다.

Sudoku [8] 방법은 다중 클래스 노드 분류 문제를 이진 노드 분류 문제로 변환해 GNN의 그래프 크기 불변성을 활용하는 접근 방식[9]을 사용했다. [10] Kakuro 문제는 반복 관계 네트워크를 사용해 해결했으며, Sudoku보다 수렴하는 데 더 많은 시간이 걸렸다.

III. 방법론

모델은 주로 그래프 컨볼루션 네트워크(GCN), 그래프 어텐션 네트워크(GAT), GraphSAGE 아키텍처를 기반으로 한다. 또한, 그래프 기반 모델이 노노그램을 해결하는 데 적합함을 증명하기 위해 CNN 및 RNN 모델의 결과와 비교한다.

데이터셋

연구에 사용된 데이터셋은 WebPbn 플랫폼에서 수집한 9400개 이상의 노노그램 퍼즐로 구성되며, 각 퍼즐은 단서, 솔루션 그리드 및 결과 이미지를 포함한다. 이 연구에서는 흑백 퍼즐만 사용하며, 데이터셋은 훈련 세트와 테스트 세트로 80%:20% 비율로 나눈다.

노드와 엣지 생성

노노그램의 구조는 그래프 형태로 자연스럽게 변환된다. 퍼즐의 각 셀은 노드로 간주되고, 인접한 셀과의 관계는 엣지로 표현된다. 기본적으로 각 셀은 상하좌우로 연결될 수 있으며, 대각선 연결도 고려하지만 이는 정확도를 감소시킨다.

임베딩

GNN 모델을 위해 행과 열 단서의 임베딩 크기를 40으로 고정한다. 단서 길이가 다른 퍼즐을 일관되게 처리하기 위해 왼쪽 제로 패딩을 사용한다. 그 후, 행과 열의 특징을 결합하여 모델의 입력으로 사용한다.

GNN 기반 모델 성능

세 가지 주요 그래프 기반 아키텍처(GCN, GAT, GraphSAGE)를 고려한다. 각 모델의 주요 목표는 주어진 단서로부터 올바른 패턴을 유추하여 그리드의 셀을 채우는 것이다.

	GCN	GAT	GraphSAGE
계층 수	3 layers	3 layers	2 layers
드롭 아웃 비율	50%	50%	50%
히든 채널 수	16	32	16
헤드 수 (GAT 전용)	-	1	-
훈련 에포크 수	30	30	30
정확도 (테스트)	64.38%	66.74%	71.26%

표 2
모델 아키텍처, 하이퍼파라미터 및 성능 개요

IV. 실험

Non-GNN 모델과의 비교

GNN의 성능을 확인하기 위해 전통적인 Non-GNN 모델(CNN[14], RNN[15])과 비교한다. CNN 모델은 세 개의 컨볼루션 레이어로 설계되며, RNN 모델은 노노그램 그리드를 시퀀스로 처리한다. 그러나 RNN 모델은 긴 시퀀스 길이로 인해 최적 이하의 결과를 보인다.

	GraphSAGE	CNN	RNN
계층 수	2	5	1
히든 채널 수	16	64	512
드롭 아웃 비율	50%	50%	50%
정확도	71.26%	11.59%	11.23%

표 3
서로 다른 모델 아키텍처와 노노그램 데이터에 대한 성능 비교 (전치 형식으로 제공됨)

CNN 성능에 미치는 크기 변동의 영향

CNN 모델의 낮은 정확도의 주요 원인이 과도한 패딩일 수 있다고 가정하고, 고정 크기(20x20)의 노노그램 데이터셋을 사용하여 실험을 수행한다. 이 실험에서 CNN 모델의 정확도는 60.92%로 크게 향상된다. 이는 다양한 크기의 노노그램을 처리하기 위한 과도한 패딩이 CNN 모델 성능에 해롭다는 점을 강조하며, 입력 크기 변동에 더 강한 아키텍처에 대한 추가 연구가 필요함을 시사한다.

제한된 모델들은 노노그램 데이터셋에서 유망한 결과를 보여주지만, 고유의 한계가 있다:

- **비교용 솔루션:** 그림 2와 같은 비교용 솔루션이 있을 수 있다.
- **2차원 제약:** 모델은 주로 2차원 노노그램 퍼즐에 초점을 맞춘다. 노노그램은 3차원 공간에서도 존재할 수 있으며, 이 경우 퍼즐의 도전과 복잡성이 크게 증가한다. 현재 아키텍처는 수정 없는 3D 노노그램에 원활하게 적용되지 않을 수 있다.
- **복잡성과 확장성:** 노노그램의 복잡성이 증가함에 따라, 특히 그리드 크기가 커질수록 모델의 계산 부담이 커진다. 더 큰 퍼즐과 복잡한 퍼즐에 대해서는 추가적인 최적화 기술이 필요하다.

이 한계점을 고려하여 제안된 방법론을 배포하거나 추가로 발전시킬 때 주의해야 한다. 향후 연구는 3차원 노노그램에 더 잘 맞는 아키텍처 개발과 문제 특유의 기능을 통합하는 데 중점을 둘 수 있다.

V. 결론

이 연구에서는 그래프 신경망(GNN)을 활용하여 노노그램 퍼즐을 해결하는 문제를 다룬다. 우리의 탐구는 퍼즐 내에 내재된 구조화된 정보를 활용하는 데 중점을 두며, 이 도메인과 관련된 복잡성과 과제를 밝혀낸다. 연구 결과, GNN은 공간적 관계를 포착하는 능력 덕분에 전통적인 Non-GNN 모델보다 노노그램 퍼즐에 더 효과적이라는 것을 확인한다. 특히 GCN, GAT, GraphSAGE 모델이 일관되게 다른 모델보다 우수한 성능을 보인다. 그러나 연구는 몇 가지 한계도 드러낸다. 2차원 GNN이 유망한 결과를 보였지만, 노노그램 퍼즐의 공간적 특성을 고려할 때 3차원 아키텍처가 성능을 더욱 향상시킬 가능성이 있다. 이는 향후 연구의 방향을 제시한다.

Non-GNN 모델과의 비교는 GNN의 유연성과 확장성을 강조하며, 특히 CNN 모델과 달리 크기 조정이나 패딩 없이 다양한 그리드 크기의 노노그램을 처리하는 데 있어 GNN이 탁월함을 확인한다.

결론적으로, 이 연구는 퍼즐 해결 분야에서 GNN의 잠재력을 강조하며, 구조화된 문제 접근에 대한 새로운 관점을 제공한다. 향후 연구로는 더 정교한 GNN 아키텍처를 탐구하고, 훈련 전략을 보강하며, 3D GNN에 대해 심도 있게 연구하는 것이 노노그램 퍼즐 해결의 한계를 더욱 넓힐 수 있을 것이다.

ACKNOWLEDGMENT

본 연구는 한국과학기술정보연구원(KISTI)의 위탁연구개발과제 (P24023)와 2020년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원 (P0012724, 2020년 산업전문인력역량강화사업)의 지원을 받아 수행된 연구임.

참고 문헌

- [1] Ueda, Nobuhisa, and Tadaaki Nagao. "NP-completeness results for NONOGRAM via parsimonious reductions." preprint (1996).
- [2] Batenburg, Kees Joost, and Walter A. Kosters. "Solving Nonograms by combining relaxations." *Pattern Recognition* 42.8 (2009): 1672-1683.
- [3] Yu, Chiung-Hsueh, Hui-Lung Lee, and Ling-Hwei Chen. "An efficient algorithm for solving nonograms." *Applied Intelligence* 35 (2011): 18-31.
- [4] Batenburg, Kees Joost, and Walter A. Kosters. "A discrete tomography approach to Japanese puzzles." *Proceedings of the 16th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC)*. 2004.
- [5] Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." (2016).
- [6] Veličković, Petar, et al. "Graph attention networks." (2017).
- [7] Hamilton, Will, Zhitaoying, and Jure Leskovec. "Inductive representation learning on large graphs." *Advances in neural information processing systems* 30 (2017).
- [8] Palm, Rasmus, Ulrich Paquet, and Ole Winther. "Recurrent relational networks." *Advances in neural information processing systems* 31 (2018).
- [9] Nandwani, Yatin, Vidit Jain, and Parag Singla. "Neural models for output-space invariance in combinatorial problems." (2022).
- [10] Daniec, Wiktor. "Solving Kakuro Problems with Recurrent Relational Networks." (2020).