

그래프 신경망에 대한 그래디언트 부스팅 기법

장은조¹, 이기용¹
¹숙명여자대학교 컴퓨터과학과
 {wkddmswh99, kiyounglee}@sookmyung.ac.kr

A Gradient Boosting Method for Graph Neural Networks

Eunjo Jang¹, Ki Yong Lee¹
¹Department of Computer Science, Sookmyung Women's University

요 약

최근 여러 분야에서 그래프 신경망(graph neural network, GNN)이 활발히 연구되고 있다. 하지만 지금까지 대부분의 GNN 연구는 단일 GNN 모델의 성능을 향상하는 데 집중되었다. 본 논문에서는 앙상블(ensemble) 기법의 대표적 기법인 그래디언트 부스팅(gradient boosting)을 이용하여 GNN의 앙상블 모델을 만드는 방법을 제안한다. 제안 방법은 앞서 만들어진 GNN의 오차를 경사 하강법(gradient descent)을 이용하여 감소시키는 방향으로 다음 GNN을 생성한다. 이 과정을 반복하여 GNN의 최종 앙상블 모델을 얻는다. 실험에서 GNN의 대표적인 모델인 그래프 합성곱 신경망(graph convolutional network, GCN)에 제안 방법을 적용하여 앙상블 모델을 생성한 결과, 단일 GCN 모델에 비해 노드 분류 정확도가 11.3%까지 증가하였음을 확인하였다.

1. 서론

그래프(graph)란 노드(node)들과 그들을 연결하는 간선(edge)들로 이루어진 데이터 구조이다. 그래프 데이터의 예시로는 페이스북과 같은 소셜 네트워크에서 사람들 간의 친구 관계, 넷플릭스와 같은 서비스에서 사용자와 영상 간의 시청 관계 등이 있다.

최근 다양한 분야에서 그래프 데이터를 분석하기 위해 인공지능망에 기반한 그래프 신경망(graph neural network, GNN)이 활발히 연구되고 있다[1]. 그래프 신경망은 그래프에 속한 각 노드의 특징을 학습하거나 그래프 전체의 특징을 학습하는 데 널리 사용된다. 하지만 지금까지 GNN에 관한 대부분의 연구는 단일 GNN 모델의 성능을 향상하는 데 집중되었다. 예를 들어 각 노드의 특징을 보다 정확히 학습할 수 있도록 하는 단일 GNN 모델의 아키텍처는 활발히 연구되었지만, GNN 모델을 여러 개 사용하여 성능을 높이는 방법은 비교적 연구가 많이 이루어지지 않았다. 하지만 단일 GNN 모델은 층(layer)의 개수가 증가할수록 모든 노드의 특징이 비슷하게 학습된다는 지나친 획일화(over-smoothing) 문제가 존재한다[2]. 이것은 GNN의 층 개수가 증가

할수록 각 노드의 특징을 학습하는데 더 멀리 떨어진 이웃 노드들의 특징까지 반영되기 때문이다. 따라서 층의 개수를 증가시켜 단일 GNN 모델의 성능을 향상하는 데는 현재로서는 한계가 존재한다.

한편 앙상블(ensemble) 기법이란 여러 개의 약한(weak) 모델들을 결합하여 단일 모델보다 성능이 높은 모델을 만드는 기법이다. 그래디언트 부스팅(gradient boosting)[3]은 앙상블 기법의 대표적인 기법으로서, 앞서 만들어진 약한 모델의 오차를 줄이는 방향으로 다음 약한 모델을 생성한다. 이 과정을 반복하여 여러 개의 약한 모델들을 생성한 후, 이들의 예측값을 모아 최종 예측을 수행함으로써 단일 모델보다 좋은 성능을 보일 수 있게 된다.

따라서 본 논문에서는 단일 GNN 모델의 한계를 극복하기 위해, 대표적인 앙상블 기법인 그래디언트 부스팅을 사용하여 GNN에 대한 앙상블 모델을 만드는 방법을 제안한다. 제안 방법은 우선 최초의 GNN을 생성한 뒤, 이 GNN의 오차를 줄이는 것을 목표로 하여 다음 GNN을 생성한다. 이때 경사 하강법(gradient descent)을 이용하여 다음 GNN의 손실 함수를 정의한다. 이 과정을 충분한 성능을 얻을 때까지 반복하여 여러 개의 GNN들로 이루어진 최종

양상블 모델을 얻는다. 따라서 제안 방법은 GNN의 층을 늘리지 않고도 여러 개의 GNN들을 양상블로 결합함으로써 단일 GNN 모델보다 더 좋은 성능을 얻을 수 있다. 실험데이터를 사용한 실험에서 제안 방법은 단일 GNN 모델에 비해 노드 분류 정확도가 11.3%p까지 증가하였음을 확인하였다.

본 논문은 다음과 같이 구성된다. 2장에서는 관련 연구를 살펴보고, 3장에서는 제안하는 GNN 양상블 기법을 설명한다. 4장에서는 성능 평가 결과를 제시하고, 마지막 5장에서는 결론을 맺는다.

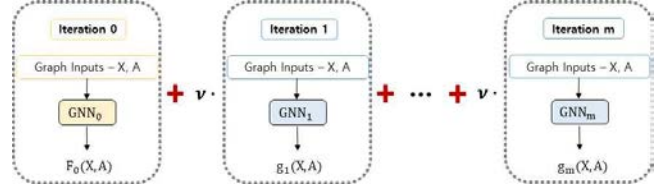
2. 관련 연구

1장에서 언급한 바와 같이 지금까지 대부분의 GNN 연구들은 단일 GNN 모델의 성능 향상에 집중되었다. 하지만 인공지능망 또는 GNN에 양상블 법을 적용한 연구도 이미 일부 진행된 바가 있으며, 본 절에서는 이들에 대해 간략히 살펴본다.

[4]는 얇은 신경망(shallow neural network)을 약한 모델로 사용하여 그래디언트 부스팅을 적용한 연구이다. 하지만 그래프 데이터는 각 노드의 특징뿐만 아니라 노드 간의 연결 관계까지 포함하기 때문에 얇은 신경망으로는 그래프 데이터의 특징을 표현하는 데 한계가 있다. [5]는 GNN의 입력인 노드들의 특징 벡터들을 그래디언트 부스팅 의사결정 트리(gradient boosted decision tree)를 사용하여 GNN의 학습에 더 효과적인 특징 벡터들로 변환하는 기법을 제안하였다. 하지만 [5]는 여러 개의 GNN을 사용하는 양상블 기법에 대한 연구가 아니라 단일 GNN 모델의 성능을 그래디언트 부스팅으로 향상시키는 기법에 대한 연구이다. [6]은 양상블 기법의 하나인 AdaBoost 기법을 사용하여 여러 개의 GNN을 생성하고 이들을 순환 신경망(recurrent neural network)과 비슷한 방식으로 연결한 구조의 단일 GNN 모델을 제안하였다. 하지만 [6]은 단일 GNN 모델의 층 개수를 성능 저하 없이 증가시키려는 연구이며, 여러 개의 독립적인 GNN을 사용하는 양상블 기법에 관한 것은 아니다.

3. 제안 방법

본 장에서는 그래디언트 부스팅을 이용하여 GNN에 대한 양상블 모델을 만드는 기법을 제안한다. 그림 1은 제안 방법의 흐름을 나타낸다. 제안 방법은 (1) 최초 GNN 생성 단계와 (2) 그래디언트 부스팅을 이용한 후속 GNN 생성 단계로 구성된다.



(그림 1) 제안 방법의 흐름도

3.1 최초 GNN 생성 단계

N 개의 노드를 가진 그래프 G 가 주어졌다고 하자. G 는 각 노드의 특징을 나타내는 특징행렬 $X \in R^{N \times F}$ 와 노드 간 연결 정보를 나타내는 인접행렬 $A \in R^{N \times N}$ 로 표현된다. 여기서 F 는 각 노드의 특징 개수를 나타낸다. GNN은 X 와 A 를 입력으로 받아 행렬 $Z \in R^{N \times K}$ 를 반환한다. Z 의 각 행은 각 노드에 대한 표현(representation) 벡터를 나타내며, K 는 각 노드에 대한 표현 벡터의 차원을 나타낸다. 예를 들어, 본 논문의 실험에서는 식 (1)으로 표현되는 2층 그래프 합성곱 신경망(graph convolutional network, GCN)[7]을 사용하였다. GCN은 이미지에 대한 합성곱을 일반화하여 그래프 데이터에 적용한 대표적인 GNN 모델이다.

$$Z = \text{softmax}(\hat{A} \cdot \text{ReLU}(\hat{A}XW^{(0)}))W^{(1)} \quad (1)$$

식 (1)에서 $\hat{A} = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$ 이며, $\tilde{A} = A + I$ 이고 \tilde{D} 는 \tilde{A} 의 차수 행렬(degree matrix)을 나타낸다. $W^{(0)} \in R^{F \times D_1}$ 와 $W^{(1)} \in R^{D_1 \times K}$ 는 각각 첫 번째와 두 번째 GCN 층의 가중치 행렬을 나타낸다.

최초 GNN을 생성할 때는 주어진 각 노드의 레이블 $Y \in R^{N \times K}$ 와 GNN이 출력한 $Z \in R^{N \times K}$ 간 오차를 최소화하는 것을 목표로 GNN을 훈련시킨다. 이를 위한 손실함수로 분류의 경우에는 크로스 엔트로피(cross entropy), 회귀의 경우에는 평균제곱오차(mean square error, MSE)를 사용할 수 있다. 이렇게 생성된 최초의 GNN을 $F_0(X, A)$ 라 나타낸다.

3.2 후속 GNN 생성 단계

제안 방법은 이전 GNN이 발생시킨 오차를 감소시키는 것을 목표로 다음 GNN을 순차적으로 생성한다. 제안 방법에서 생성하는 양상블 모델은 다음 식으로 나타낼 수 있다.

$$F_m(X, A) = F_{m-1}(X, A) + \nu \cdot g_m(X, A) \quad (2)$$

식 (2)에서 $F_m(X, A)$ 은 m 번째 반복에서 생성된, 총 $(m + 1)$ 개의 GNN으로 이루어진 양상블 모델을

나타내며, ν 는 학습률(learning rate)을, $g_m(X,A)$ 은 m 번째 반복에서 새로 생성된 GNN을 나타낸다. 제안 방법은 m 번째 반복에서 다음 식을 만족하는 $g_m(X,A)$ 를 생성한다.

$$g_m(X,A) = \arg \min_{g(X,A)} \left[-\frac{\partial(F_{m-1}(X,A) - Y)^2}{\partial F_{m-1}(X,A)} - g(X,A) \right] \quad (3)$$

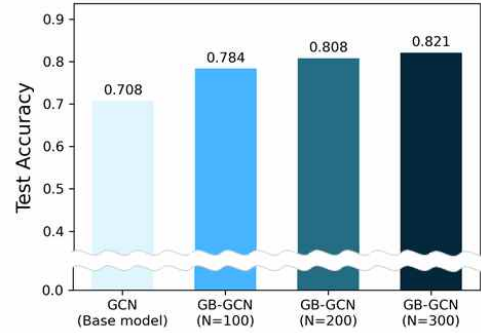
식 (3)에서 $-\partial(F_{m-1}(X,A) - Y)^2/\partial F_{m-1}(X,A)$ 는 경사 하강법(gradient descent)을 사용하여 $F_{m-1}(X,A)$ 와 Y 간의 오차를 감소시키는 방향을 근사한 값을 나타내며, $g_m(X,A)$ 는 이 값과의 차이를 최소화하는 것을 목표로 학습된다. 제안 방법은 $F_m(X,A)$ 의 성능이 충분히 향상될 때까지 식 (3)에 따라 $g_1(X,A)$, $g_2(X,A)$, ..., $g_m(X,A)$ 를 순차적으로 생성한다.

4. 실험 결과

본 실험에서는 단일 GCN 모델의 성능과 제안 방법으로 생성한 GNN 앙상블 모델의 성능을 비교하였다. 단일 GCN 모델과 GNN 앙상블 모델의 약한 모델로는 모두 식(1)이 나타내는 2층 GCN을 사용하였으며, 성능 척도로는 각 모델이 그래프의 각 노드를 분류하는 정확도를 측정하였다. 실험에 사용된 모델들은 모두 PyTorch로 구현하였다.

실험 데이터로는 과학 출판물 간의 인용 관계를 나타내는 Cora 데이터셋[8]을 사용하였다. Cora 데이터셋은 2,708개의 노드와 5,429개의 간선을 가진 그래프로 표현되며, 각 노드는 1,433개의 특징을 가지고 있으며 7개의 클래스 중 하나로 분류된다. 훈련 데이터, 검증 데이터, 테스트 데이터로는 각각 1,760, 407, 541개의 노드 데이터를 사용하였다. 2층 GCN의 은닉층 노드는 1,433개로 설정하였으며, 제안 방법의 식(2)에서 사용하는 매개변수 ν 의 값은 0.01로 설정하였다.

그림 2는 단일 GCN 모델(GCN)과 제안 방법으로 생성한 앙상블 모델(GB-GCN)의 성능을 비교한 결과이다. N은 앙상블 모델에 사용된 GCN의 개수를 나타낸다. 그림 2에서 제안 방법은 N이 100, 200, 300개로 증가함에 따라 단일 GCN 모델에 비해 성능이 11.3%p까지 향상되는 것을 볼 수 있다. 이것은 제안 방법이 그래디언트 부스팅 방법에 기반하여 이전 모델의 오차를 보완하는 방식으로 여러 개의 GCN을 생성하여 이들을 결합하기 때문이다. 따라서 제안 방법은 단일 GCN 모델의 성능을 더 이상 향상시키기 어려울 때 효과적으로 사용될 수 있다.



(그림 2) 제안 방법의 성능 평가 결과

5. 결론

본 논문은 GNN에 대한 그래디언트 부스팅 기반의 새로운 앙상블 기법을 제안하였다. 제안 방법은 앞서 생성된 GNN의 오차를 경사 하강법을 사용하여 감소시키는 방향으로 다음 GNN을 반복적으로 생성한다. 실제 데이터를 사용한 실험을 통해 제안 방법은 단일 GCN 모델에 비해 노드 분류 정확도를 최대 11.3%p까지 향상함을 확인하였다. 제안 방법은 실험에서 사용한 GCN 뿐만 아니라 GAT, AGNN, APPNP, AGNN 등 다른 GNN에 대해서도 일반적으로 적용할 수 있다.

Acknowledgement

이 성과는 2021년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. NRF-2021R1A2C1012543).

참고문헌

[1] Z. Wu et al, "A comprehensive survey on graph neural networks," IEEE Transactions on Neural Networks and Learning Systems, 32(1), pp. 4-24, 2020.
 [2] Y. Yang et al, "Graph neural networks inspired by classical iterative algorithms," ICML, 2021.
 [3] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," Annals of statistics, pp. 1189-1232, 2001.
 [4] S. Badirli et al., "Gradient Boosting Neural Networks: GrowNet," arXiv: 2002.07971, 2020.
 [5] S. Ivanov and L. Prokhorenkova, "Boost then convolve: Gradient boosting meets graph neural networks", ICLR, 2021.
 [6] K. Sun et al., "AdaGCN: Adaboosting graph convolutional networks into deep models," ICLR, 2021.
 [7] T. N. Kipf and M. Welling., "Semi-supervised classification with graph convolutional networks," arXiv:1609.02907, 2016.
 [8] Cora, <https://paperswithcode.com/dataset/cora>.