**Higher-Order Explanations of Graph Neural Networks via Relevant Walks**

<https://arxiv.org/pdf/2006.03589.pdf>

**0. INTRODUCTION**

|  |
| --- |
| **GNN (Graph Neural Network):** 일반적인 그래프 구조에서 관측을 통해 학습하는 방법   * GNN이 유용한 예측을 생성한다고 해도 **일반적으로 black-box**이다. * 따라서, 학습된 모델에서 **새로운 insight를 직접 추출**할 수도 없고, model이 그래프 구조에서 **의도한 대로 작동한다는 것을 보장**할 수도 없다.   여기서는 **GNN의 예측에 대한 XAI method**를 제안한다.   * 여기서는 GNN에 의해 구현된 함수는 **입력 그래프에 대해 지역적으로 다항식(polynomial)**으로 작동한다는 것을 이용한다.   다음 그림은 GNN-LRP를 나타낸 것이다. 여기서 **설명 과정은 GNN의 output에서 시작**하고, **예측과 관련된 walk을 uncover**하는 방법으로 **backward 방향**으로 진행한다. |

**1. TOWARDS EXPLAINING GNNS**

|  |
| --- |
| GNN은 **graph 자료구조를 input으로 하는 특별한 구조의 Neural Network**이다.   * Graph는 directed/undirected, labeled/unlabeled, spatial, time-evolving과 같은 **다양한 form**을 가질 수 있다. * 그래프 자료구조의 **높은 이질성(heterogeneity)을 handling**하기 위해 수많은 종류의 GNN이 개발되었다. * 이들 GNN의 공통점은 **input graph가 first layer에 위치하지 않지만, 여러 개의 layer에서 등장**한다는 것이다. * Graph Neural Network는 **몇 개의 interaction block을 쌓는 방법**으로 구성되며, **각 블럭 t=1,…,T는 graph representation 를 계산**한다. |
| **NEXT PAGE** |

|  |
| --- |
| **PREVIOUS PAGE** |
| **하나의 block 내부에서 representation**은 다음의 step을 적용하여 생성된다.   |  |  | | --- | --- | | Aggregate | 각 node가 **이웃한 node로부터 정보를 수신**한다. | | Combine | 각 node에서 **새로운 feature를 추출**한다. |  * 이들 두 step은 연속된 블럭의 representation **과 를 연결**한다.   여기서 **는 크기가 n\*n인 matrix 형태로 주어진 input graph**이다.   * **:** node K와 연관된 의 행, **:** combine 함수 (일반적으로 one-layer 또는 multi-layer neural network으로 **그래프의 각 node로부터 새로운 representation을 생성**) * GNN에 대한 전체 input-output relation은 다음의 함수로 표현된다.   **GNN이 initial state에 대한 함수**라고 볼 때 다음과 같이 생각할 수 있다.  이때 어떤 **reference point** 에서의 **함수 f에 대한 Taylor expansion**은 다음과 같다. |

**2. EXPLAINING GNNS IN PRACTICE**

|  |
| --- |
| High-order Taylor expansion이 간단히 제시되었으나, **시스템적으로 High-order 미분값을 신경망에서 찾는 것은 어렵고** complex model에 적용하기 어렵다.   * 이 한계점을 극복하기 위해 여기서는 **walk W라는 개념을 적용**한다. * **Walk W:** **GNN의 연속적인 layer**에서 node를 연결하는 **edge의 ordered sequence** |
| **NEXT PAGE** |

|  |
| --- |
| **PREVIOUS PAGE** |
| 각 walk은 특정한 bag-of-edge에 mapping되기 때문에, **walk-based explanation은 bag-of-edges explanation의 모든 정보**를 갖고 있다.   * 특히 walk-based explanation으로부터 **항상 bag-of-edges explanation을 복구**할 수 있다.  |  | | --- | | 1. **walk-based explanation**은 GNN의 여러 layer가 prediction에 도달하는 것과 같은 방법으로 정보를 제공한다.   * Figure 2에서처럼 **두 node 사이에 message passing**이 이루어졌는지 확인할 수 있게 한다.   2. **GNN의 구조에 더 직접적으로 연결된 walk**은 **explanation을 계산**하는 데 더 많은 실질적인 이익을 가져다줄 수 있다. |   알고리즘을 보다 간단히 보이기 위해서, 새로운 변수 를 도입한다. 이것은 **GNN의 서로 다른 layer에서 발생하는 edge를 구분**하기 위한 것이다.   * GNN **output**을 **expanded input에 대한 함수, 즉** 로 나타낸다. * **Node-based notation**을 추가로 적용한다. 이 notation에서는 **walk들이 first layer에서 top layer로 traverse하는 node의 sequence**로 주어진다. (예시: )   **[The GNN-GI Baseline]**  GNN-GI는 다음의 수식이 각각의 walk으로 decompose될 수 있다는 것에 착안했다. 또한, **T-order derivative는 first-order derivative로 변환**될 수 있으며, 이는 GNN에서 지역적으로 정의된다.  그러나, **GNN-GI에는 다음과 같은 한계점**이 있다.   * 네트워크가 **positive homogeneous가 아닌** 경우, conservation property 는 더 이상 만족되지 않는다. * 예측의 **상당한 부분이 input variable까지 전달되는 데 실패**한다. * Deep model에서는 GI method가 의존하는 **gradient는 shattering effect에 영향**을 받으며, 이것은 noise를 만든다. |
| **NEXT PAGE** |

|  |
| --- |
| **PREVIOUS PAGE** |
| **[Better Explanations with GNN-LRP]**  GNN-GI baseline의 한계를 극복하기 위해, **Layer-wise Relevance Propagation (LRP)**를 적용하고, 이것을 GNN을 설명하는 데 적용한다.   * LRP는 **Deep Neural Network**에 특화된 설명 기술이다. * LRP는 DNN의 output을 **top layer에서부터 input layer까지 propagate**하고, 각 레이어에서 propagation rule이 적용된다.   GNN-LRP는 GNN-GI의 발전된 방법으로, **GI의 attribution step을 LRP의 attribution step으로 변환**한다.   |  | | --- | | **<GI의 attribution step>** | | **<LRP의 attribution step>** |   **Nested LRP procedure**를 그림으로 나타내면 다음과 같다.    **Neuron k에 도달하는 relevance**의 계산 과정은 다음과 같다. (단, 는 대응되는 neuron activation)   |  | | --- | | **Step 1.** 이라고 하자. 여기서 는 locally approximately constant term이다.  **Step 2.** 의 mapping을 정의한다. 여기서 **는 상수 값**으로 간주한다. 그 후 **relevance score 을 계산하기 위해 LRP를 적용**한다.  **Step 3.** Pooling 을 적용하고, **그 결과가 로 쓰일 수 있음**을 보인다. 여기서 은 locally approximately constant이다. |  * 이것이 성립할 때, 이것은 **LRP의 모든 GNN (from top-layer to first layer)에 대한 응용**을 가능하게 한다. |
| **NEXT PAGE** |

**<서로 다른 종류의 GNN에 대한 GNN-LRP의 propagation rule>**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Aggregate | Combine | GNN-LRP Rule |
| GCN  **[29]** |  |  |  |
| GIN  **[39]** |  |  |  |
| Spectral **[38], [40]** |  |  |  |

|  |
| --- |
| **: neuron j와 neuron k를 연결하는 행렬**이고, 그 행렬의 element를 라고 한다.  **:** **rectification** function  **:** 여기서 는 propagation rule의 hyperparameter이다.  **:** **indicator function** (Dirac delta) 함수로, **j와 k가 각각의 node에서 같은 위치의 neuron**일 때 1이다.  **:** **[38]** 에 나타나 있는 **graph convolutional filter approximation**의 1개의 component |

|  |
| --- |
| **PREVIOUS PAGE** |
| GNN-LRP를 구현하기 위해서, 여기서는 **forward-rewrite strategy를 적용하여 편리하고 간결하게 구현**한다. 여기서는 GCN의 case에 대해서만 논하지만, **실제로는 다른 GNN 모델로 확장**하여 적용할 수 있다.   * 여기서는 relevance score를 **대응되는 activation과 몇몇 factor들의 곱**으로 계산한다. 즉, , 등으로 나타낼 수 있다. 또한, 다음과 같이 간결하게 나타낼 수 있다.   이 형태는 **다음과 같은 gradient form**과 유사하다.  GCN의 GNN-LRP rule에서 **combine step을 다음과 같이 재정의**한다.  여기서 과 는 각각 **element-wise multiplication과 element-wise division**을 의미한다. |
| **NEXT PAGE** |

|  |
| --- |
| **PREVIOUS PAGE** |
| 또한, forward pass가 **walk W에 대해 mask가 선택되는 방법**으로 각 layer에 대해 한번 rewritten되면, 다음과 같이 relevance score를 계산할 수 있다. |

**3. TESTING AND VALIDATING GNN-LRP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 제안된 GNN-LRP method를 테스트하기 위해, **몇 가지 graph prediction task에 대해 많은 종류의 GNN을 학습**시킨다. 이때 class 구성은 다음과 같다.   |  |  | | --- | --- | | First class | Barabasi-Albert graphs, of **growth parameter 1** (BA-1) | | Second class | **Slightly higher growth model**, and new nodes attached preferably to low-degree nodes |   (왼쪽: **synthetic dataset의 2가지 class**의 데이터의 일부,  오른쪽: class 1의 그래프의 **GIN prediction에 대한 GNN-LRP 설명**)    **RED:**  Relevant (positively contributing) walks  **BLUE:**  Negatively contributing walks  **CIRCLES:**  Stationary walk/walks  서로 같은 그래프에 대한 **서로 다른 explanation technique**의 비교 |
| **NEXT PAGE** |

|  |
| --- |
| **PREVIOUS PAGE** |
| **[Quantitative Comparison]**  GNN-LRP가 시스템적으로 우수한 explanation을 생성한다는 것을 보장하기 위해 양적 평가를 진행한다. 이를 위해 **image classifier를 위한 일반적인 evaluation technique인 pixel-flipping을 이용**한다.  **Subgraph의 relevance**는 각각의 attribution method에 대해 다음과 같이 적용된다.   * 여기서 는 subgraph, 는 그 subgraph의 예측에 대한 contribution이다. * 따라서 **node를 추가하거나 제거하는 것이 GNN output에 미치는 효과**는 로 나타낼 수 있다.  |  |  | | --- | --- | | Node-based attribution | **특정한 subgraph** 에 대한 relevance는 **각 node의 contribution의 합**으로 계산된다. | | Edge-based attribution | Relevance는 **각 edge 에 대한 합**으로 주어진다. | | Higher-order attribution | Explanation의 단위가 bags of edges이며, 이것은 수식으로 계산된다. (for **GNN-LRP** and **GNN-GI**)   * 이때 membership relation 는 **bag 에 있는 모든 edge가 subgraph 에 속해야** 한다는 것을 의미한다. |   다음의 2가지 **node-flipping task**를 고려한다.   |  |  | | --- | --- | | Activation task | 그래프에 node를 추가하여 **GNN output을 최대한 빠르게 최대화**하는 것   * 일반적으로 node의 최적의 sequence를 찾는 것은 복잡하다. * **주어진 subgraph 에 대해 Greedy approach**를 적용하여 해결할 수 있다. 즉, 다음에 추가할 best node는 다음과 같다. | | Pruning task | Original graph 에서 시작해서, **GNN output이 가장 적게 영향을 받는 node를 그래프에서 제거**한다. 여기서 제거하기 가장 좋은 node는…   * 이 방법을 이용하면, **GNN-GI와 GNN-LRP와 같은 high-order technique들의 성능이 저평가**될 수 있다. | |
| **NEXT PAGE** |

|  |
| --- |
| **PREVIOUS PAGE** |
| node flipping procedure와 그에 대한 **AUFC**는 다음 그림과 같다.    **Activation Task와 Pruning Task에서의 AUFC score**는 각각 다음과 같다. |

**4. NEW INSIGHTS WITH GNN-LRP**

|  |
| --- |
| **1. Sentiment Analysis**  Natural Language Processing (NLP) 에서는 text data가 sequence로 프로세싱되며, 이에 대응되는 **문법 구조는 parse tree**라고 할 수 있다. 이 중 **후자는 학습 알고리즘에 대한 추가적인 입력**으로 주어질 수 있다. |
| **NEXT PAGE** |

|  |
| --- |
| **PREVIOUS PAGE** |
| **2. Quantum Chemistry**  화학에 적용되는 머신러닝에서 GNN은 **분자 구조를 예측하는 데 높은 성능**을 보이고 있다.   * 여기서 목표는 **분자 내부의 어떤 구조가 분자의 에너지에 긍정적인/부정적인 영향을 주는지**에 대한 insight를 얻는 것이다. * GNN-LRP에 의해 생성된 Walk에 기반한 설명이 매우 자세하기 때문에, 그 중 일부는 단일 분자에서 보이지 않는다. 따라서 우리는 **GNN-LRP 설명이 여러 가지 형태로 보이는** 것에 대한 실험을 진행한다.       **3. Image Classification**  Convolutional Neural Network은 pixel에 대한 격자에 대해 작동하는 **특정한 Graph Neural Network (GNN)**으로 볼 수 있다. |