**Software for Dataset-wide XAI: From Local Explanations to Global Insights with Zennit, CoRelAy, and ViRelAy**

[**https://arxiv.org/pdf/2106.13200.pdf**](https://arxiv.org/pdf/2106.13200.pdf)

1. **Overview**

|  |
| --- |
| 이 논문에서는 **Dataset-wide XAI**에 기반한 machine learning model들의 reasoning을 탐색하기 위한 소프트웨어 패키지에 대해 알아본다.   * **Zennit:** PyTorch를 이용한, customize가 가능한 local XAI 프레임워크이다.   + **LRP (Layer-wise Relevance Propagation)과 같은 Rule-based approach**와 PyTorch의 Module 구조에 초점을 맞춘다. * **CoRelAy:** 정교하고 dataset-wide한 분석 파이프라인을 빠르게 build하는 데 사용된다.   + 이것은 **processing, clustering 및 embedding step**을 포함한다.   + 분석 및 재사용에서 **효율성**을 중시한다. * **ViRelAy:** Zennit과 CoRelAy의 분석 결과에 대해, **사용자에게 친숙한 entry point**를 제공한다. (interactive web-application 형태)   + Model attribution, clustering 및 visualizable embedding에 대해 탐색하는 동안, 연구자는 동료들에게 특정한 발견을 공유할 수 있다.   이 도구들을 결합하면, 이 3가지 도구들은 XAI가 거대한 모델 및 데이터의 양적 및 질적인 탐색을 가능하게 한다.   * Local model explanation에서는 **Zennit**을 사용한다. * 사용자는 전체 dataset에 대해 계산된 거대한 attribution set을 **CoRelAy**를 통해 build된 파이프라인을 통해 분석할 수 있다. * 이 결과를 **ViRelAy**를 통해 시각화한다. |

1. **Attribution with Zennit**

|  |
| --- |
| **Zennit은 PyTorch를 이용한 attribution framework을 제공하며, 이것은 PyTorch의 Module 구조에 기반하고, Autograd 및 Hook 기능을 사용한다.**   * LRP에서 사용되는 Rule-based 접근 방법의 단순하고 직관적인 구현에 초점을 맞춘다. * 간단한 attribution method인 **SmoothGrad**와 **Integrated Gradient** 등도 구현되었다. (rule-based system을 사용하지는 않지만, 분석될 모델의 gradient의 straight-forward function임) |
| **(Cont.)** |

1. **Attribution with Zennit (Cont.)**

|  |
| --- |
| **<Rule-Based Attributions>**  Zennit에서는 **rule-based attribution**들이 **forward and backword Hooks를 Modules (layers)에 attach**함으로써 계산된다.   * Model의 gradient를 계산하는 것은 그 대신 desired attribution을 제공한다. * Rules는 **BasicHook에 함수를 제공**하여 **general attribution method를 inputs, parameters, accumulators**에 대해 커스터마이징할 수 있다.   **<Mapping Rules with Composites>**  Rule-based attribution을 다룰 때 가장 큰 문제점은 **desired rule들을 모든 개별적인 레이어에 assign**하는 것이다.   * Zennit는 이것을 **Composite를 구현**하여 해결하며, 이것은 **Module-property를 rule로 매핑**한 것이다.   + Module-property들은 예를 들어 function의 종류, function의 파라미터, 하이퍼파라미터 등이 있다. * Built-in된 기본적인 composite의 한 가지 예시는 SpecialFirstLayerMapComposite로, 레이어의 종류에 기반하여 rule을 assign한다.   **<Temporary Model Modification with Canonizers>**  Rule-based attribution method의 다른 문제점은 이들 규칙이 canonical form으로 변환되지 않는 한, **많은 네트워크에 직접적으로 적용되지 않는다**는 것이다.   * 예를 들어, 가장 끝부분에 단 1개의 activationa만 존재하는 여러 개의 연속된 linear layer는 LRP의 모든 variant에 trivial하게 기인할 수 없다.   + 이러한 구조의 대표적인 예시는 batch normalization이다.   **<Attributors>**  Attributor는 다음을 수행하는 optional한 convenience function이다.   * 주어진 모델과 Composite에 대한 **gradient를 계산** * SmoothGrad 또는 Integrated Gradient와 같은 **black-box attribution 접근 방식 구현**   주어진 **gradient-based black-box attribution 접근 방식** (Smooth-Grad 등)에 대하여, **LRP와 SmoothGrad와 같은 조합**을 계산하기 위한 Composite를 공급하는 것은 항상 가능하다.  **<Heatmaps>**  이미지 데이터에 대한 attribution은 일반적으로 heatmap으로 표시되는데, Zennit은 **attribution을 heatmap image로 쉽게 시각화하고 저장**할 수 있는 이미지 모듈을 포함한다.   * 이미지는 **기인된 relevance와 대응되는 index**에 대해, intensity와 8-bit palette를 이용하여 저장된다. |

1. **Building Analysis Pipelines with CoRelAy**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Attribution method들이 모델의 예측 전략에 대한 양적인 insight를 줄 때, 사용자는 개별적인 heatmap의 attribution들이 어떻게 모델의 추론이 되는지를 추측한다.**   * **모델에 대한 더 깊은 insight는 dataset-wise 분석을 통해 얻어진다.**     (Heatmaps of attributions of lighthouses)  **Spectral Relevance Analysis:** 모델의 예측 전략을 시각적으로 임베딩하고 attribution들을 **Spectral Clustering**과 **t-distributed Stochastic Neighborhood Embedding (t-SNE)**으로 임베딩하여 양적으로 예측한다.   * CoRelAy는 SpRAy와 같이 **양적인 분석 파이프라인**을 생성하는 것이다. * 반면, CoRelAy의 main use-case와 motivation은 **Zennit에 의해 생성된 attribution을 분석**하는 것이다. * CoRelAy는 또한 multiple clustering 및 embedding을 위한 빠른 dataset 탐색을 위해 사용될 수 있다.   **<Processors and Params>**   |  |  | | --- | --- | | Processors | **Pipeline 안에 있는 action**   * 구현하기 위해 inheriting class에서 function이라는 이름의 method를 구현해야 한다. | | Params | **서로 인접한 instance**에 기반하여 바뀌는 **descriptor** | | Pipeline | Params 중 is\_output은 **Processor의 output이 Pipeline에 의해 반환**되어야 한다는 signal이다. (intermediate인 경우 포함) | | Parallel | 이전 Processor의 output은 **여러 개의 Processor로 전달**되어야 한다. | |
| **(Cont.)** |

1. **Building Analysis Pipelines with CoRelAy (Cont.)**

|  |
| --- |
| **<Pipelines and Tasks>**  Pipeline은 feed-forward 함수로, **파이프라인 실행을 위해 front에서 back까지 수행되어야 하는 Task**를 가지고 있다.   * CoRelAy에서는 Pipeline은 **computation template**으로 보일 수 있으며, 특정한 결과를 계산하기 위해 몇 가지 step이 involve되어 있다. * Task는 **default Processor와 허용된 종류의 Processor를 포함**(optional)하는 이러한 step이다. |

1. **Interactive Visualization with ViRelAy**

|  |
| --- |
| **양적 분석에서는 많은 양의 결과가 도출되며, 서로 다른 결과를 연결하고 원본 데이터와 함께 표현하는 것은 어렵다.**   * SpRAy에 의해 수행되는 분석은 다음과 같은 object를 포함한다.   + Source **data points**   + **Attributions** of these source data points (=model)   + Visual **2-dimensional representation** of embedded attribution data   + **Clustering labels**   + **Global auxiliary scores** * ViRelAy는 이들 결과가 **5가지의 object들을 시각적으로 연결함**으로써 탐색되게 하는 웹 어플리케이션이다.   **<실행 화면>** |
| **(Cont.)** |

1. **Interactive Visualization with ViRelAy (Cont.)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **<Data Loading>**  **ViRelAy는 CoRelAy의 데이터를 처리**하기 위해 생성되었으며, CoRelAy의 데이터는 **ViRelAy가 post-hoc을 사용할 수 있는 구조**로 HDF5 파일에 저장된다.   * Source data와 attribution data에 대한 분석 파일은 각각 HDF5에 적용되며, project file에서 참조된다. * 서로 다른 dataset 또는 attribution **접근 방법을 비교**하기 위하여, ViRelAy는 **임의의 용량의 프로젝트 파일**을 공급할 수 있다.  |  |  | | --- | --- | | 1 | Project file로 dictate된 project에 대한 **project selection** | | 2 | **Analysis selection** (analysis 방법, 카테고리, 클러스터링 방법, 임베딩을 선택 가능) | | 3 | **Color map selection** | | 4 | Data/attribution **시각화 모드** 선택 | | 5 | **Import and export** 버튼으로, 현재 선택된 analysis, 카테고리, 클러스터링, 임베딩, color map, 시각화 모드를 export할 수 있음 | | 6 | Data point에 대한 시각화 | | 7 | Auxiliary category score plot | | 8 | Cluster point selection | | 9 | **Data/attribution visualization** ((4)에서 선택한 시각화 모드에 따라 source data, attribution Heatmap 등을 표시) | |