2024 전기 졸업과제 착수보고서

질병 연관 유전자 발굴을 위한 머신러닝 기법 설계

7조 서울대구부산

목차

1. 과제 배경 및 목표

a. 과제 배경

인간의 유전자에서 질병과의 관계를 파악하는 것은 질병의 발병가능성을 예측하고 초기에 대응하기 위해 중요한 생물학적 문제이다. 여러 임상 실험을 통해 유전자와 질병 간의 관계를 파악하거나 유전자 데이터 분석을 통해 유전자와 질병의 관계를 밝혀내기도 한다. 하지만 한 유전자가 여러 질병에 거쳐 발현되거나 인간의 전체 유전자 수 중 확인된 유전자가 한정되어 있어서 이러한 방식은 한계가 존재한다. 이를 해결하기 비교적 최근 연구에서는 유전자 간의 유사성을 통해 질병과의 연관을 예측하는 방식을 도입했다. 즉, 기존에 알려진 질병 유전자와 유사한 특성을 가지는 후보 유전자는 질병과 관련될 가능성이 크다는 것을 의미한다.

질병 연관 유전자를 예측하기 위한 모델로 Collaborative Filtering, Matrix Factorization, Graph Convolution Network를 사용하는 경우가 있다. 유전자와 질병은 복잡한 고차원 관계를 가지고 있으므로 이를 반영할 수 있는 그래프 구조의 GCN을 적용해 예측 모델을 생성할 계획이다.

b. 과제 목표

본 과제에서는 GCN 모델을 통해 질병과 유전자, 유전자와 유전자의 관계를 학습하고 이를 통해 후보 유전자가 질병과의 연관성이 존재하는지 파악하는 것을 목표로 한다.

2. 진행 방안

a. 데이터 전처리

I. DisGeNET

DIsGeNET은 인간 질병과 관련된 유전자에 대한 정보를 제공하며 인간 질병의 기초적 조사, 질병 유전자 분석, 약물 치료 효과 등 다양한 의료 연구 목적으로 사용되는 플랫폼이다. DisGeNET에서는 21671개의 유전자와 30170개의 질병, 장애, 비정상적 표현 간의 연관성 데이터를 제공하고 유전자 변이와 질병 간의 연관성 데이터도 제공한다. 본 과제에서는 DisGeNET에서 제공하는 질병 연관성 데이터, 유전자 연관성 데이터를 이용해 질병과 유전자의 연관성을 예측할 계획이다.

	A	B	C	D	E	F	G
1	diseaseId	diseaseName	diseaseType	diseaseClass	diseaseSemanticType	NofGenes	NofPmids
2	C0000727	Abdomen, Acute	phenotype	C23	Sign or Symptom	2	2
3	C0000729	Abdominal Cramps	phenotype	C16	Sign or Symptom	1	1
4	C0000731	Abdomen distended	phenotype	C06	Finding	103	0
5	C0000734	Abdominal mass	phenotype	C06	Finding	2	0

[disease association]

	A	B	C	D	E	F	G	H	I
1	geneId	geneSymbol	DSI	DPI	PLI	protein_class_name	protein_class	NofDiseases	NofPmids
2	1	A1BG	0.7	0.538	4.99E-09	Receptor	DTO_05007575	27	20
3	2	A2M	0.529	0.769	4.52E-11	Enzyme modulator	DTO_05007584	147	145
4	3	A2MP1						1	1
5	9	NAT1	0.536	0.846	1.93E-14	Enzyme	DTO_05007624	133	184

[gene association]

II. PCA(Principal Component Analysis)

PCA는 주성분 분석 이라고도 하며 고차원의 데이터를 분석하기 위한 기법으로 차원 축소(Dimensionality Reduction)를 위해 사용된다. 원래 데이터의 분산을 최대한 보존하는 새로운 축을 찾고, 그 축에 데이터를 사영시키는 방식으로 차원을 축소한다. PCA를 통해 원래 데이터에 대한 정보 손실을 최대한 줄이면서 차원을 축소 할 수 있다.

b. 개발환경

개발 언어 : Python

개발 **OS** : Window

사용 라이브러리 : PyTorch

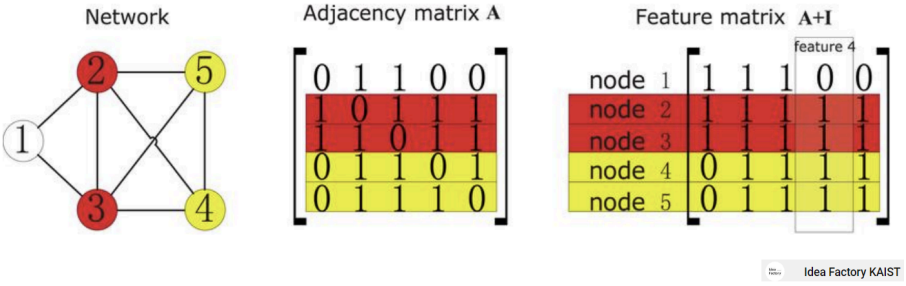
사용 플랫폼 : Jupyter Notebook, Google Colab

c. 주요기술

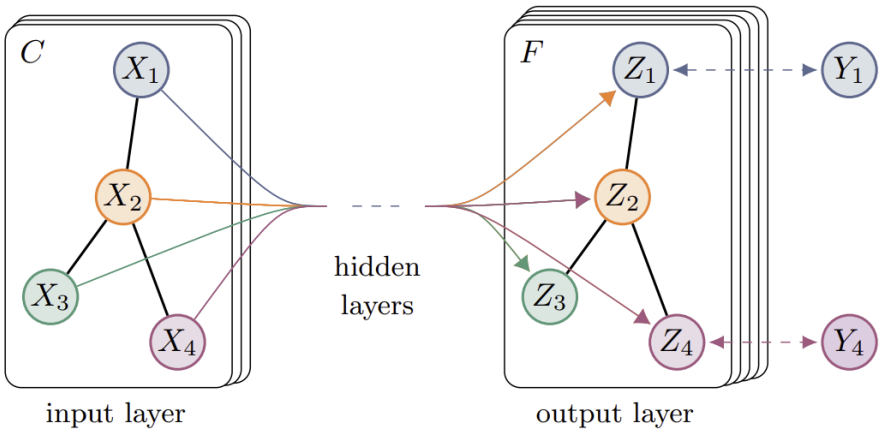
I. GCN - Graph Convolutional Network

Graph - 정점(Vertex)과 정점을 연결하는 간선(Edge)으로 구성된 구조로, 간선의 특징에 따라 여러 종류로 나눌 수 있다.

- node : 각각의 노드마다 피처벡터(노드의 정보)를 갖는다.
- edge : 인접행렬 (Adjacency matrix)로 표현이 된다.
  - 노드간의 연결을 1과 0으로 나타낸다. (n x n 행렬)



GCN - Graph Convolution Network



Graph Convolutional Network (GCN)는 Graph 형태의 데이터에서 Convolution을 통해 유의미한 정보(Feature)를 뽑아내는 Neural Network이다. GCN는 그래프 구조를 통해 각 노드의 특징을 효과적으로 학습하며, 주로 반지도 학습(Semi-supervised learning)에 사용된다. GCN의 입력은 다음과 같은 행렬들로 구성된다.

Input Matrix

1) Feature matrix

   모든 노드의 Feature vector를 모아둔 행렬이다. $X \in R^{NxF}$ 형태로 N은 노드의 수, F는 각 노드의 특징 수를 나타낸다. 따라서 하나의 행이 한 노드의 Feature vector를 나타낸다.

2) Adjacency Matrix

   노드 간의 연결 여부를 나타내는 행렬로서 그래프의 구조를 표현한다. A_{ij}가 1이면 노드 i와 노드 j가 연결됨을 나타낸다. 이전 상태의 정보를 유지하기 위해 대각선 값(diagonal)을 모두 1로 설정한다.

GCN의 각 layer에서 노드는 Convolution의 과정을 거쳐 Input Matrix를 변환하게 되는데 layer를 거칠수록 하나의 노드에는 더 많은 Feature 정보를 갖게된다. 이때 row의 개수(노드의 수)는 일정하지만, filter에 따라 column의 수(Feature의 수)가 달라질 수 있다. 그래프의 형태(Edge, Node의 구성)은 바뀌지 않지만, Feature matrix의 값(Node의 정보)은 각 layer를 통과하면서 업데이트 된다.

반지도 학습(Semi-supervised learning)은 label이 있는 데이터와 없는 데이터를 함께 사용하여 모델을 학습시키는 방식으로, 상대적으로 극히 희소하게 파악된 Gene-Disease 정보를 통해 아직 알려지지 않은 수많은 Gene-Disease 관계 탐색에 활용된다.

Label이 있는 데이터는 손실 함수 (cross-entropy)를 사용하여 예측된 label과 실제 label 간의 차이를 최소화하면 GCN 레이어를 통해 그래프 전체로 전파되어 Label이 없는 데이터는의 embedding 학습에도 기여한다.

II. PyTorch

PyTorch는 딥러닝 구현을 위한 파이썬 기반의 오픈소스 머신러닝 라이브러리로 단순한 선형 회귀 알고리즘부터 복잡한 처리 작업에 사용되는 생성형 트랜스포머 모델까지 다양한 아키텍처를 지원한다. 또한 CUDA와 같은 API를 통해 GPU를 연산에 이용할 수 있고 모델 학습을 최적화하고 가속화할 수 있다. 따라서 GCN 모델의 틀을 구성하고 학습 과정을 최적화하기 위해 PyTorch 라이브러리를 사용할 예정이다.

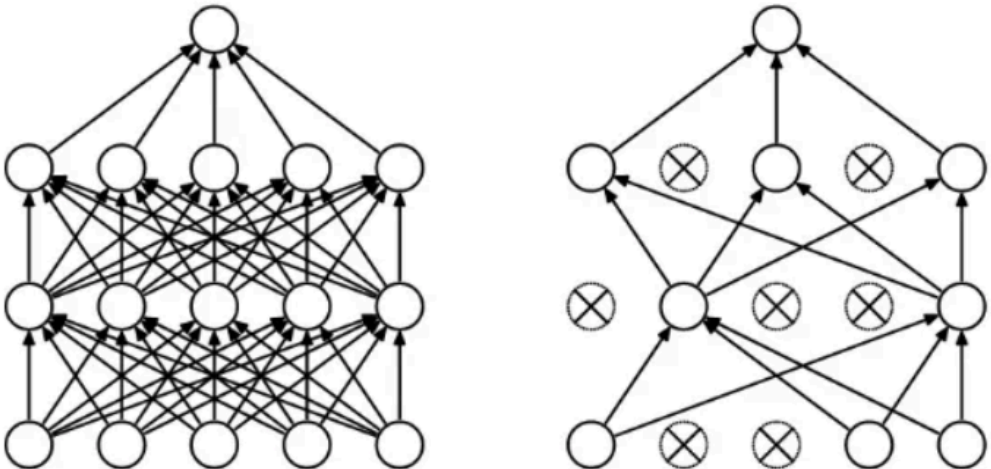
d. 최적화 방법

I. Overfitting & Underfitting

Overfitting(과적합)은 모델이 학습데이터에 과하게 적합한 상태이거나 정확하게 일치할때 발생한다. 따라서 모델이 학습 데이터가 아닌 새로운 데이터에 대해서는 정확한 예측을 생성하거나 결론을 도출 할 수 없게 된다. Underfitting(과소적합)은 학습 데이터도 학습을 하지 못한 상태를 의미한다. 언더피팅이 일어나는 이유로는 학습 반복 횟수가 너무 적은, 데이터의 양이 너무 적음 등이 있다. 과적합과 과소적합이 일어나지 않도록 하기 위해서는 적절한 모델과 데이터를 선택하고, 모델과 훈련에 대한 적절한 하이퍼 파라미터를 찾아야한다.

II. Drop out

신경망에서 확률적으로 일부 노드를 학습에서 제외하는 방식으로 모델 학습 과정에서의 과적합을 막기 위한 기법이다. 이와 더불어 Mini-batch learning을 통해 전체 데이터를 작은 batch 단위로 나누어 학습을 진행하면 일부 feature의 과적합을 막고 여러 데이터 셋의 feature을 반영할 수 있도록 학습을 진행할 수 있다. Drop out의 확률



p는 층마다 다르게 설정할 수 있으며, 일반적으로 입력층에 가까울수록 낮은 값이 사용된다.

3. 개발 일정 및 역할 분담

a. 개발 일정

개발 구분	추진 내용	추진 일정 (월별, 상/하 구분)											
		5월		6월		7월		8월		9월		10월	
		上	下	上	下	上	下	上	下	上	下	上	下
계획	착수보고서 작성	■	■										
분석	주요 기술 이해		■	■	■								
	선례 연구 이해 및 분석		■	■	■								
설계	데이터탐색				■	■							
	개발 환경 구축				■	■							
	학습 모델 기법 연구				■	■	■						
개발	데이터셋 전처리					■	■	■					
	베이스모델 구축					■	■	■					
	중간보고서 작성						■	■					
	모델 학습 및 클리닝						■	■	■	■	■		
테스트	모델 테스트 및 최적화								■	■	■	■	
마무리	최종보고서 작성										■	■	■
	결과물 업로드 및 후속 처리												■

b. 조원별 담당 역할

조원	역할 분담
김이경	개발 환경 선정 및 구축, 데이터 전처리
박화성	데이터 전처리, 모델 성능 평가
이윤재	Hyperparameter 조정 및 모델 최적화
전체	이전 연구 분석, GCN 학습, 데이터 셋 수집, 학습 모델 구축, 보고서 작성

4. 참고자료

[1] Peng Han, Peng Yang, Peilin Zhao, Shuo Shang, Yong Liu, Jiayu Zhou, Xin Gao, Panos Kalnis, GCN-MF: Disease-Gene Association Identification By Graph Convolutional Networks and Matrix Factorization, 2019

[2] Thomas N. Kipf, Max Welling, Semi-Supervised Classification with Graph Convolutional Networks

[3] https://github.com/heartcored98/Standalone-DeepLearning/tree/master

[4] https://www.disgenet.org

[5] https://kh-kim.github.io/nlp_with_deep_learning_blog/docs/1-14-regularizations/04-dropout/

[6] https://builtin.com/data-science/step-step-explanation-principal-component-analysis

[7] https://www.broadinstitute.org/news/new-maps-link-thousands-genetic-variants-disease-genes