Disease-Gene Interaction Prediction with Graph Neural Networks

1. Bevezető (Introduction)

A betegségek és gének közötti kapcsolatok feltárása a bioinformatika és orvosi kutatások egyik kulcsfontosságú területe. Ezen kapcsolatok megértése hozzájárulhat új diagnosztikai eszközök és kezelési módszerek kidolgozásához. A DisGeNET adatbázis az egyik legátfogóbb forrás, amely információt szolgáltat a betegségek és gének közötti kapcsolatokról. Ez az adatbázis összegyűjti és integrálja a tudományos irodalomból, adatbázisokból és elemzésekből származó információkat, lehetővé téve az átfogó vizsgálatokat.

A projekt célja egy grafikus neurális hálózat (GNN) alkalmazása a betegségek és gének közötti kapcsolatok előrejelzésére. A GNN-ek képesek a komplex, kapcsolati struktúrák kihasználására, és hatékonyan modellezik a grafikus adatokban rejlő információt. Ezáltal az algoritmus nemcsak az ismert kapcsolatokat elemzi, hanem képes új, potenciális kapcsolatok azonosítására is, hozzájárulva az orvosi kutatások és kezelési stratégiák előrelépéséhez.

Kapcsolódó munkák

Számos tudományos kutatás foglalkozik a grafikus neurális hálózatok bioinformatikai alkalmazásaival:

1. **Kipf et al. (2016): Graph Convolutional Networks**
    - A GCN-ek bevezetése és alapvető működési elveinek bemutatása. A szerzők részletezik, hogyan lehet a csomópontok és azok kapcsolatai alapján hatékonyan osztályozási feladatokat megoldani.
    - Hivatkozás az arXiv-ra
2. **Hamilton et al. (2017): GraphSAGE**
    - A GraphSAGE algoritmus bemutatása, amely a csomópontok reprezentációját azok lokális környezete alapján tanulja meg. Ez a módszer hatékonyan kezel nagyobb gráfokat is.
    - Hivatkozás az arXiv-ra
3. **DisGeNET adatbázis**

- A DisGeNET API dokumentációja és az adatok feldolgozásának lehetőségei. Ez az adatbázis integrált információt nyújt betegég-gén kapcsolatokról, amelyek a jelen projekt alapját képezik.
- DisGeNET weboldal

4. **PyTorch Geometric**
   - Egy hatékony eszköz grafikus neurális hálózatok implementálására. A könyvtár rugalmas megoldásokat nyújt a gráf-alapú mélytanulási modellekhez.
   - GitHub repository

Ez a projekt a fent említett forrásokra épít, és átfogó képet nyújt a grafikus neurális hálózatok bioinformatikai alkalmazásáról.

LLM használata, alkalmazása

A projekt során a nagyméretű nyelvi modelleket (LLM) az alábbiakra használtam:

- GitHub Copilot és OpenAI ChatGPT segítségével:
  - Kódgenerálás: Python kódok megírásához és az adatszerkezetek feldolgozásához.
  - Kommentek generálása: A kódrészek érthetőségének növelése.
- OpenAI ChatGPT segítségével:
  - Dokumentáció szövegének generálása: Az egyes fejezetek tartalmának kidolgozásához.
  - Stilisztikai és helyesírási javítások: A dokumentum összhangjának és olvashatóságának biztosítására.

Az LLM-ek által generált tartalmakat minden esetben kipróbáltam, és csak azután építettem be a projektbe, miután a várt eredményt hozta.

2. Módszerek (Methods)

Adatok előkészítése (Data Preparation):

- A DisGeNET API-ból lekért adatok feldolgozása:
  - **Disease Entity API**: Betegségazonosítók kinyerése (pl. MONDO, UMLS).

  - **GDA Summary API**: Betegség-gén kapcsolatok lekérdezése és mentése CSV formátumban.

- A gráf adatszerkezet létrehozása:
  - **Csomópontok (Nodes):** Betegségek és gének.
  - **Élek (Edges):** Betegség-gén kapcsolatok, súlyozva a score értékkel (ha van).

Modell tervezése (Model Design):

- **Baseline modell**:
  - Logisztikus regresszió/döntési fa használata kiindulási pontként.
  - Egyszerűség és gyors implementáció.
- **GNN modell**:
  - **Hálózat típusa:** Graph Convolutional Network (GCN).
  - **Miért ezt választottuk:** Képes a gráfstruktúra relációs információinak kihasználására.
- **Hálózat architektúra:**
  - Bemeneti réteg: csomópont jellemzők.
  - Konvolúciós rétegek: gráfon végzett aggregáció (GCN).
  - Kimeneti réteg: kapcsolat valószínűsége (link prediction).

Hiperparaméter optimalizálás (Hyperparameter Optimization):

- **Kézi tuning:** Tanulási ráta, rétegek száma, rejtett dimenziók mérete.
- **Automatizált keresés:** Optuna vagy Grid Search alkalmazása (ha történt).

3. Kiértékelés (Evaluation)

- **Teszt adatok eredményei:**
  - Osztályozás esetén:

        - Pontosság (accuracy), F1-score, érzékenység (recall), specifikusság (specificity).
        - Tévesztési mátrix (confusion matrix) bemutatása.
    - Regresszió esetén:
        - Regplot: valódi vs. prediktált értékek vizualizálása.
- **Vizualizációk:**
    - ROC-görbék és AUC-értékek (ha releváns).
    - Gráfstruktúra vizualizáció (csomópontok és élek megjelenítése).

4. Következtetések (Conclusions)

- **Fő eredmények:**
    - Hogyan teljesített a GNN a baseline modellhez képest?
    - Milyen kapcsolatokat sikerült előre jelezni?
- **Tapasztalatok:**
    - Mi működött jól és mi nem?
    - Milyen javításokat lehetne eszközölni a jövőben?
- **Jövőbeli fejlesztési lehetőségek:**
    - Adatbővítés más forrásokkal.
    - Fejlettebb modellek, pl. Graph Attention Networks (GAT).