Dokumentáció a GCN Edge Prediction projekthez

Bevezető

A projekt célja egy személyre szabott barátajánló rendszer kifejlesztése Graph Neural Networks (GNNs) segítségével, amely a Facebook Ego Network adathalmazán alapul. Az ilyen rendszerek célja, hogy a hálózatban lévő kapcsolati mintázatokat elemezve új, potenciális baráti kapcsolatokat javasoljanak a felhasználóknak. A gráf neurális hálózatok (GNN) különösen hatékony eszközök a hálózati adatok elemzésében, mivel képesek megragadni a gráfstruktúrákban rejlő információkat, például a csúcsok lokális környezetét és a hálózat globális szerkezetét.

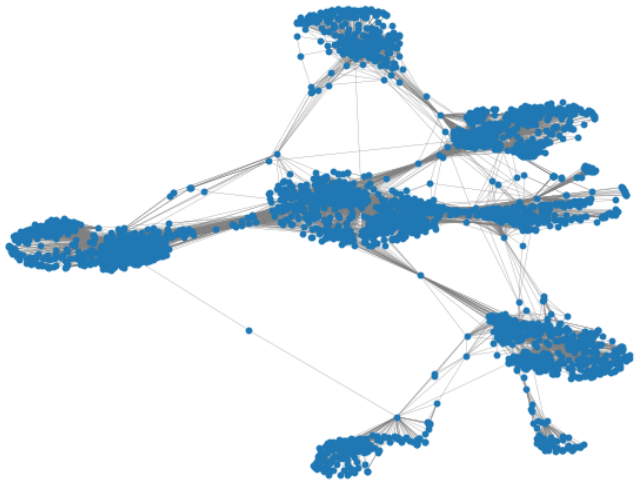
Az adathalmaz a Stanford SNAP adatkészletéből származik, és a Facebook felhasználói kapcsolatok egy kis, ego-alapú részhalmazát tartalmazza. A gráf csúcsai egyes személyeket képviselnek, míg az élek azokat a kapcsolatokat jelölik, amelyek közöttük fennállnak. A projekt során ezeket az adatokat elemeztük, előkészítettük és egy GCN-alapú modellt fejlesztettünk az élpredikció problémájának megoldására.

Több tudományos cikket is áttanulmányoztunk. Például a [1] alapján rájöttünk, hogy lehetséges a csúcsok klaszterizációja. A [2] alapján megismertük a GCN-eket. Ezeken kívül tanulmányoztuk a [4] és a [5] cikkeket.

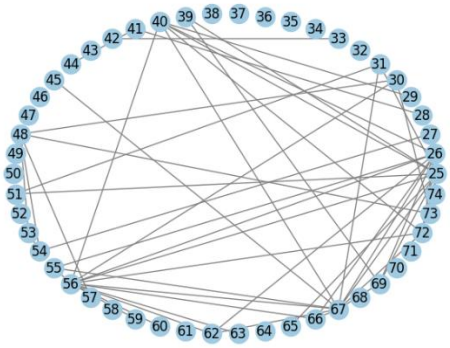
Adatok előkészítése és elemzése

A projekt első lépése az adathalmaz betöltése és elemzése volt. A Facebook Ego Network adathalmaz [3] letöltésével kezdtük a feladatot. Az adathalmaz az éleket tartalmazza olyan módon, hogy minden sorban két darab szám szerepel, a gráf két csúcsa, amik között él fut.

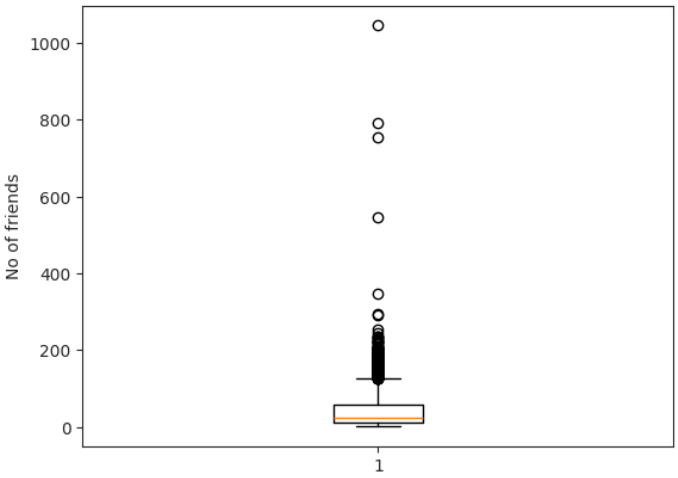
Ez után a gráfot a NetworkX könyvtár segítségével olvastuk be, amely lehetővé tette a gráf éleinek és csúcsainak manipulációját. A gráfot a következő ábra mutatja, ami az adathalmazban található összes élet és csúcsot tartalmazza még az adattisztítás előtt. A kék pontok jelölik a gráf csúcsait, míg a szürke vonalak a gráf élei. Ekkor 4039 ember adatai voltak az adathalmazban 88234 éllel.



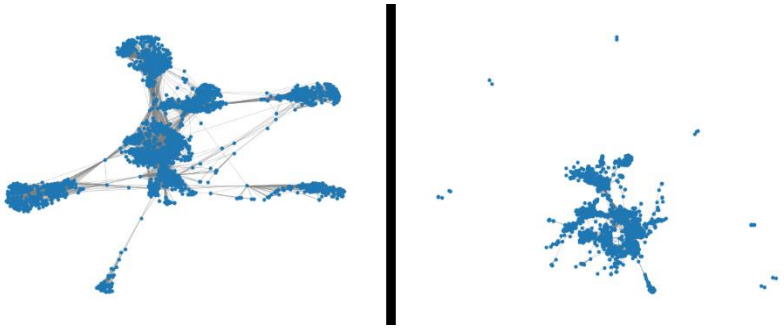
Az adatok előzetes vizsgálatához elkészítettünk egy részgráfot az eredeti hálózatból, amely 50 csúcsot és a köztük futó éleket tartalmazta. Ezt a részgráfot egy körkörös elrendezéssel vizualizáltuk, amely lehetővé tette, hogy áttekintsük a hálózat struktúráját és megértsük a kapcsolatok mintázatait. Ez látható az alábbi ábrán.



Az eredeti gráf teljes vizsgálatát követően a csúcsok fokszámának (degree) eloszlását is elemeztük, ami fontos lépés volt az outlierek és kiugró értékek azonosításában. A fokszámok eloszlását histogramként és boxplotként ábrázoltuk, amely rámutatott a gráf néhány csúcsának extrém magas fokszámára. A boxplot-ot mutatja a következő ábra.



Ezt követően eltávolítottuk azokat a csúcsokat, amelyek fokszáma kettőnél kisebb volt, hogy a modell edzését ne befolyásolják az ilyen szélsőséges értékek. Kipróbáltuk a tanítást a legnagyobb fokszámú csomópontok elhagyásával is Igaz, hogy jobb eredményt tudott elérni a háló, de ezt túl nagy változtatásnak éreztük az eredeti feladathoz képest. A következő ábra mutatja, ezt a két állapotot. A baloldalinál csak a kettőnél alacsonyabb fokszámú csúcsok lettek eltávolítva (88159 él), míg a jobboldalinál a legnagyobb fokszámú egy százaléknyi csúcs is (77147 él).



A tisztított adathalmazból az éleket pozitív mintaként használtuk fel, majd véletlenszerű negatív éleket generáltunk olyan csúcspárokból, amelyek között nem létezett él. Ez a negatív minták létrehozása fontos lépés volt, mivel a predikciós probléma lényege az élek meglétének vagy hiányának előrejelzése. A negatív minták biztosították a modell számára a kiegyensúlyozott tanulási környezetet.

Jellemzők előállítása

A gráf csúcsaihoz tartozó jellemzők előállítása kulcsfontosságú lépés volt a modell tanításához. Az alapvető csúcsjellemzők közé tartozott a fokszám (degree), amely a csúcs közvetlen szomszédainak számát jelzi, valamint az átlagos klaszterezési együttható (average clustering coefficient), amely a csúcs környezetének sűrűségét méri. Ezeket a jellemzőket Tensor formátumban tároltuk, hogy a PyTorch Geometric által elvárt bemeneti formátumnak megfeleljenek.

További jellemzőként alkalmaztuk a Node2Vec algoritmust, amely a gráf csúcsainak mélyebb reprezentációját biztosította véletlenszerű séták segítségével. A Node2Vec a lokális és globális gráfstruktúrák közötti összefüggéseket megragadva generált beágyazásokat, amelyeket tensorok formájában illesztettünk a meglévő csúcsjellemzők mellé.

A csúcsjellemzőket a gráf átmérőjével (diameter) is kiegészítettük, amely egy globális tulajdonság, és minden csúcsra ugyanazt az értéket rendeltük hozzá. Ezzel biztosítottuk, hogy a hálózat globális struktúrájának információi is elérhetők legyenek a modell számára.

Modell kialakítása

A modell fő eleme egy több rétegű gráf neurális hálózat (Graph Neural Network, GNN), amelyet kifejezetten az élpredikció problémájának megoldására terveztünk. A hálózat felépítése során figyelembe vettük, hogy a gráfstruktúrák és a csúcsok környezetében rejlő információk hatékony aggregációját biztosítsuk. A modellt úgy alakítottuk ki, hogy képes legyen egyszerre figyelembe venni a helyi (lokális) és globális kapcsolati mintázatokat, miközben a gráf geometriai sajátosságait is megragadja.

Bemeneti jellemzők transzformációja

A modell első lépése a csúcsjellemzők lineáris transzformációja volt, amelyet egy teljesen kapcsolt (fully connected) réteg valósított meg. Ez a lépés azért volt szükséges, hogy a különböző forrásokból származó csúcsjellemzők (pl. fokszám, klaszterezési együttható, Node2Vec beágyazások) egységes dimenzióba kerüljenek, és illeszkedjenek a rejtett rétegek bemeneti követelményeihez. A transzformáció után a bemeneti tensor egy standardizált jellemzőhalmazzá vált, amely a gráf minden csúcsához egyedi reprezentációt társított.

GCN rétegek és aggregáció

A gráf szomszédsági információinak aggregációját Graph Convolutional Network (GCNConv) rétegek valósították meg. A GCN rétegek a gráf szomszédsági mátrixát használták a csúcsok környezetének információinak összesítésére. Az aggregáció során minden csúcs saját jellemzői mellett a közvetlen szomszédainak jellemzőiből is tanult, ami biztosította, hogy a gráf lokális szerkezete explicit módon beépüljön a modell predikciójába. Az ilyen típusú rétegek alapvető fontosságúak voltak, mert hatékonyan tudták modellezni a hálózat topológiáját, miközben a számítási komplexitás alacsony maradt.

Továbbfejlesztett rétegek: GATConv és SAGEConv

A GCNConv rétegek mellett további rétegeket is alkalmaztunk a modellben, hogy a gráfstruktúrák különböző aspektusait is megragadhassuk. A Graph Attention Network (GATConv) réteg figyelemmechanizmust (attention) alkalmazott, amely lehetővé tette, hogy a modell differenciált súlyozással kezelje a szomszédos csúcsokat. Ez különösen akkor bizonyult hasznosnak, amikor bizonyos szomszédos csúcsok információja fontosabb volt, mint másoké. A SAGEConv réteg ezzel szemben hatékony aggregációs mechanizmust biztosított, amely nagyobb gráfok esetén is jól skálázódott.

A különböző rétegek közötti kombináció lehetővé tette, hogy a modell egyszerre ragadja meg a gráf finomabb, lokális mintázatait és a globális struktúrákat. Ez különösen fontos volt az élpredikció szempontjából, mivel az élek meglétének valószínűsége gyakran függ mindkét aspektustól.

Batch normálás és dropout

A modell stabilitásának és generalizációs képességeinek növelése érdekében minden réteg után batch normálást és dropoutot alkalmaztunk. A batch normálás standardizálta a rétegek kimenetét, ami stabilabb és gyorsabb tanulást eredményezett. A dropout rétegek véletlenszerűen kikapcsolták a neurális kapcsolatokat, ami segített elkerülni az overfittinget, különösen a kis adathalmazon történő tanítás során. Az optimális dropout arányt az Optuna keretrendszerrel végzett hiperparaméter-optimalizálás során határoztuk meg.

Residual connection-ek

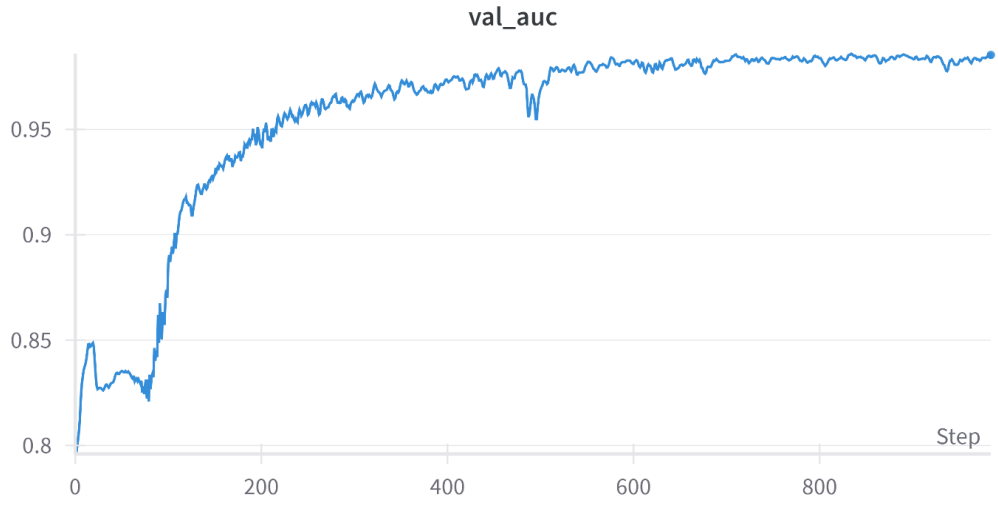
Az aggregációs rétegek között residual connection-eket alkalmaztunk, amelyek lehetővé tették, hogy a modell megőrizze az eredeti bemeneti jellemzőket is a tanulási folyamat során. Ez különösen fontos volt a mélyebb hálózatok esetében, ahol a gradiens elhalványulás problémája jelentkezhet. A residual connection-ök biztosították, hogy a tanulás hatékony legyen, miközben a különböző rétegek által tanult információk integritása is megmaradt.

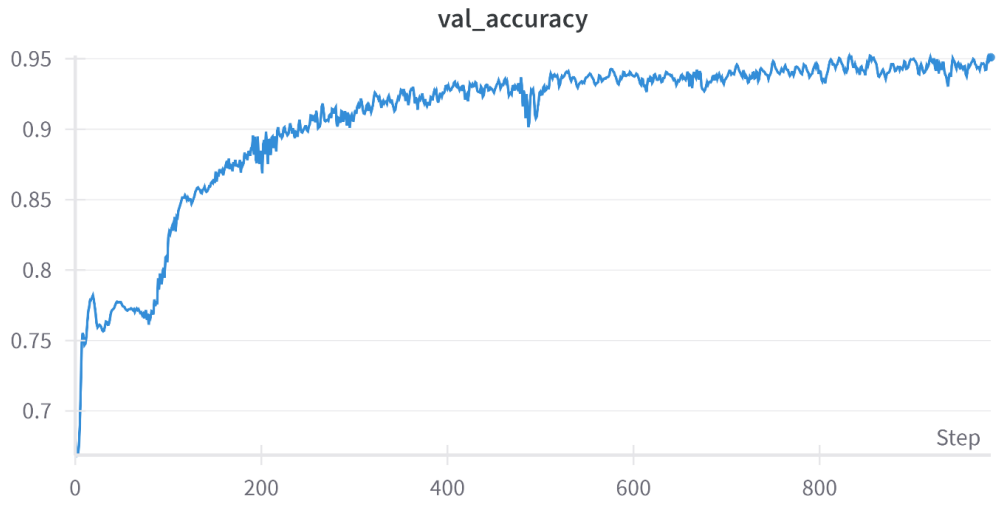
Kimeneti rétegek

Az élpredikcióhoz a modell utolsó lépése egy több rétegből álló, teljesen kapcsolt hálózat volt, amely az élekhez tartozó valószínűségeket számította ki. Az élhez tartozó jellemzők előállítása úgy történt, hogy az adott él két csúcsához tartozó jellemzőket összefűztük, majd ezeket a teljesen kapcsolt rétegeken vezettük keresztül. Az utolsó réteg egy szigmoid aktivációs függvényt alkalmazott, amely az élek meglétének valószínűségét adta meg (0 és 1 közötti érték).

Hálózat dimenziói és skálázása

A modell dimenzióinak meghatározása és a rejtett rétegek mélységének optimalizálása szintén kulcsfontosságú volt. A bemeneti csúcsjellemzők dimenzióját a tisztított adathalmaz alapján határoztuk meg. Az optimális rejtett dimenziókat és a rétegek számát az Optuna segítségével végeztük el, amely lehetővé tette, hogy a modell a legjobb teljesítményt érje el a validációs adatokon.





Kiértékelés

Teljesítmény a teszt adatokon

A modell kiértékelése során a teszt adatokon elért teljesítményt több metrikával is vizsgáltuk. Az élpredikciós probléma egy bináris osztályozási feladat, ezért az értékeléshez a pontosságot (accuracy), a ROC AUC-t, valamint a tévesztési mátrixot használtuk, hogy átfogó képet kapjunk a modell predikcióinak

minőségéről. Az osztályozás során a pozitív minták az adathalmazban meglévő éleket, míg a negatív minták a véletlenszerűen generált, nem létező éleket jelentették.

Eredmények

- **Pontosság (Accuracy):** A modell pontossága a teszt adatokon 95.107%-ot ért el, ami azt jelzi, hogy a predikciók több mint 95%-a helyes volt. Ez az eredmény különösen jó, figyelembe véve, hogy a gráfstruktúrák modellezése és az élek predikciója komplex feladat.
- **ROC AUC:** Az ROC görbe alatti terület (Area Under Curve) 0.98534 volt, ami a modell kimagasló képességét jelzi a pozitív és negatív minták megkülönböztetésére. Ez az érték azt mutatja, hogy a modell rendkívül jól általánosít, és képes pontosan rangsorolni a kapcsolati valószínűségeket.

Konklúzió

A projekt során kifejlesztett GCN-alapú élpredikciós modell kiváló eredményeket ért el, bizonyítva a gráf neurális hálók hatékonyságát a hálózati kapcsolatok előrejelzésében. Az alkalmazott GCNConv, GATConv és SAGEConv rétegek kombinációja lehetővé tette, hogy a modell egyszerre használja a gráf lokális és globális információit, ami kulcsfontosságú a pontos predikciókhoz. A modellezési folyamat során a residual connection-ek, a batch normálás és a dropout technikák mind hozzájárultak ahhoz, hogy a tanulás stabil és a modell generalizációs képessége magas legyen.

Az eredmények, különösen a 0.98534-es ROC AUC, azt mutatják, hogy a modell jól teljesít a meglévő élek és a hiányzó kapcsolatok azonosításában. A tévesztési mátrix elemzése alapján a modell jobban felismeri a meglévő kapcsolatokat, míg a hamis pozitív és hamis negatív arányok további finomhangolás lehetőségét jelzik. A regplot vizualizáció alapján a modell által generált valószínűségek szoros korrelációt mutattak a valós kapcsolatokkal, ami tovább erősíti a modell megbízhatóságát.

Jövőbeli fejlesztési lehetőségek

1. **Hálózati rétegek finomítása:** A GCN és GATConv rétegek mellett további, például Graph Isomorphism Networks (GIN) vagy Graph Transformer architektúrák tesztelése.
2. **Feature engineering:** További csúcs- és éljellemzők bevezetése, például központiság, modularitás vagy más gráfstatikai mutatók.
3. **Heterogén gráfok kezelése:** A modell adaptálása olyan hálózatokra, ahol különböző típusú csúcsok és élek is jelen vannak.
4. **Valós idejű predikció:** A modell optimalizálása nagyobb gráfokon való gyors predikcióra, például streaming adatok esetén.

A projekt eredményei alapján a kifejlesztett GCN-alapú megközelítés erős alapot biztosít a barátajánló rendszerek fejlesztéséhez és más gráf alapú predikciós problémák megoldásához. Az elért eredmények és a kiépített módszertan jelentős hozzájárulást nyújtanak a gráf neurális hálók gyakorlati alkalmazásainak bővítéséhez.

LLM-ek használata a feladathoz

A feladathoz használtuk a ChatGPT-t és a Gemini-t. Ezekkel kódot generáltunk, kommenteztünk illetve szöveget generáltunk.

A tanulmányozott irodalom jegyzéke:

[1] https://github.com/miladfa7/Social-Network-Analysis-in-Python

[2] https://rendazhang.medium.com/graph-neural-network-series-2-convolution-on-graphs-delving-into-graph-convolutional-networks-79b42b042f53

[3] https://snap.stanford.edu/data/ego-Facebook.html

[4] https://arxiv.org/abs/1607.00653

[5] https://arxiv.org/abs/1611.07308