

GIN - raport

Patryk Krukowski
Marcin Świątkowski

Data Science, semestr nr 2

January 2023

Spis treści

1	Wstęp	1
2	Zarys teoretyczny	3
2.1	Graph Isomorphism Network (GIN)	3
2.2	LDP	5
3	Eksperymenty	5
3.1	Wyniki	5
3.2	Wnioski	9
4	Zawartość załącznika	10
5	Bibliografia	10

1 Wstęp

Naszym zadaniem było zbadanie jakości działania modelu typu GIN ([Xu+18]) w praktyce. Zbiory danych (dane grafowe), na których uczyliśmy ten model po-

chodzą z repozytorium TUDataset (<https://chrsmrrs.github.io/datasets/docs/datasets/>):

- PROTEINS - zbiór danych składający się z protein, które są klasyfikowane jako enzymy lub nie. Węzły grafu reprezentują aminokwasy i są połączone, jeśli są oddalone od siebie o mniej niż 6 angstromów. Sam zbiór danych to 1113 grafów, 2 klasy (enzym lub nie), średnia liczba wierzchołków to 39.06, a średnia liczba krawędzi to 72.82.
- MUTAG - jest zbiorem związków nitroaromatycznych, a celem jest przewidywanie ich mutagenności na Salmonella typhimurium. Grafy wejściowe są używane do reprezentacji związków chemicznych, gdzie wierzchołki oznaczają atomy i są etykietowane przez typ atomu (reprezentowany przez kodowanie one-hot), podczas gdy krawędzie między wierzchołkami reprezentują wiązania między odpowiednimi atomami. Zawiera on 188 próbek związków chemicznych. Zawiera 188 grafów, 2 klasy, średnią liczbę węzłów równą 17.93, średnią liczbę krawędzi równą 19.79.
- PTC_MM - składa się z ponad trzystu cząsteczek organicznych oznaczonych według ich rakotwórczości na samcach i samicach myszy i szczurów. Jest w nim 336 grafów, 2 klasy, średnio 13.97 węzłów i 14.32 krawędzi.
- ENZYMES - jest zbiorem 600 struktur trzeciorzędowych (grafy) białek uzyskanych z enzymatycznej bazy danych BRENDA. Zbiór danych ENZYMES zawiera 6 enzymów (klasy). Ponadto znajduje się w nim średnio 32.63 węzły oraz 62.14 wierzchołków.
- DD - jest oparty na tych samych danych, co zbiór danych PROTEINS, ale zawiera grafy, w których węzły reprezentują poszczególne aminokwasy, a krawędzie ich przestrzenne sąsiedztwo. Zawiera 1178 grafów, 2 klasy, średnio 284.32 węzły i średnio 715.66 krawędzi.
- NCI1 - pochodzi z dziedziny cheminformatyki, gdzie każdy graf wejściowy jest używany jako reprezentacja związku chemicznego: każdy wierzchołek oznacza atom cząsteczki, a krawędzie między wierzchołkami reprezentują wiązania między atomami. Ten zbiór danych jest związany z ekranami przeciwnowotworowymi, gdzie substancje chemiczne są oceniane jako

pozytywne lub negatywne dla raka płuc. Każdy wierzchołek ma etykietę wejściową reprezentującą odpowiadający mu typ atomu, zakodowany przez schemat one-hot-encoding do wektora elementów 0/1. Składa się z 4410 grafów, dwóch klas, średnio 29.87 wierzchołków 32.30 krawędzi.

Przejdziemy teraz do przedstawienia teorii stojącej za GIN oraz za LDP, pełniącym rolę baseline'u. Następnie przedstawimy wyniki nauki wspomnianych modeli oraz przedstawimy krzywe uczenia, których wartości zostały uśrednione w oknie czasowym równym 10 jednostek w celu wyeliminowania szumu obecnego w danych.

Warto także wspomnieć, że udało nam się uruchomić uczenie modeli lokalnie na GPU, korzystając z oprogramowania CUDA.

2 Zarys teoretyczny

2.1 Graph Isomorphism Network (GIN)

Architektura GIN powstała po to, aby zmaksymalizować dyskryminatywną moc grafowych sieci neuronowych. Sama idea grafów izomorficznych polega na tym, że tak naprawdę interesuje nas **struktura** grafu z dokładnością do permutacji węzłów grafu. Chcemy utożsamiać ze sobą tego typu grafy, ponieważ niosą one tę samą wiedzę.

By "wykrywać" grafy izomorficzne, można użyć tzw. testu WL ([HV21]), który potrafi orzec, że dane 2 grafy **nie są** izomorficzne. Natomiast test ten nie może określić, czy dane dwa grafy są **na pewno** izomorficzne. W [Xu+18] udowodniono zostało, że sieci GIN są tak samo dobre, jak test WL, ponieważ w ich architekturze zaaplikowana została sieć neuronowa MLP mogąca nauczyć się iniektywnych (różnowartościowych) funkcji, co przekłada się na to, że dla dwóch grafów nieizomorficznych wyprodukowane zostaną różne osadzenia wierzchołków. Z tego powodu często możemy spotkać określenie, że GINy to najpotężniejsze grafowe sieci neuronowe. Tłumaczy to również słowo "izomorficzne" w samej nazwie.

Powyższe rozważania uwypuklają także różnicę pomiędzy grafowymi sie-

ciami neuronowymi z atencją (GAT) a sieciami typu GIN. Mianowicie w sieci GAT uczymy się odpowiedniego ważenia połączeń pomiędzy poszczególnymi węzłami, a w sieci GIN uczymy się bezpośrednio aproksymacji pewnej funkcji różnowartościowej produkującej embeddingi.

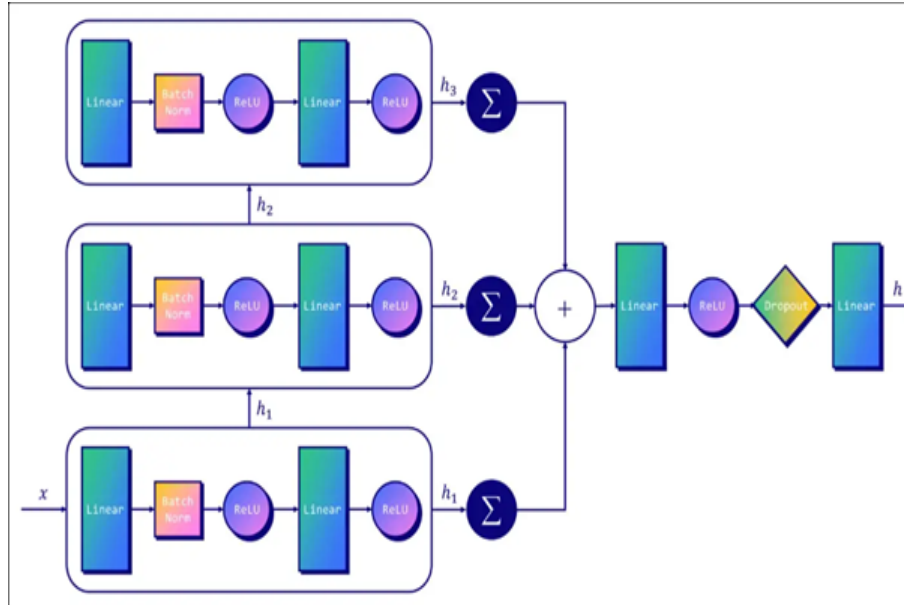
Wektor ukryty policzony dla i -tego wierzchołka liczymy w sieci GIN następująco:

$$h_i = MLP((1 + \epsilon) \cdot x_i + \sum_{j \in N_i} x_j),$$

gdzie parametr ϵ oznacza ważność danego wierzchołka i może być parametrem, którego się uczymy lub może też być ustalony "na sztywno". Natomiast x_i to wartość wierzchołka, a N_i to zbiór indeksów sąsiadów tego wierzchołka.

Mając wyznaczone wektory ukryte, możemy wyprodukować embeddingi, wykonując pooling i tutaj jako pooling stosujemy średnią, maksimum lub minimum po wektorach ukrytych (jest to tzw. readout).

Architektura sieci GIN z artykułu [Xu+18] prezentuje się następująco:



Rysunek 1: Architektura sieci GIN

I my przyjęliśmy taką architekturę. Użyliśmy też jej zmodyfikowanej wersji,

która polega na dodaniu warstw atencyjnych. Całość została zaimplementowana przy użyciu frameworku PyTorch.

2.2 LDP

Jest to baseline dla zadań klasyfikacji grafów [CW18]. Nazwa wywodzi się od **Local Degree Profile**. Opiszemy teraz pokrótce tę metodę.

Dla każdego wierzchołka v w grafie G ekstrahujemy następujące pięć cech:

- stopień wierzchołka v ,
- minimalny stopień wierzchołka spośród sąsiadów wierzchołka v ,
- maksymalny stopień wierzchołka spośród sąsiadów wierzchołka v ,
- średni stopień wierzchołków sąsiadów wierzchołka v ,
- odchylenie standardowe ze stopniów sąsiadów wierzchołka v .

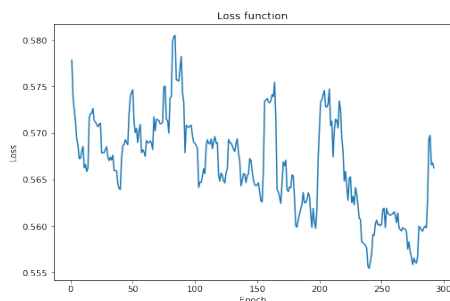
Następnie dla każdego wektora cech dla danego wierzchołka dla każdej jego cechy (współrzędnej) aplikujemy funkcję EDF (dystrybuanta empiryczna). Wektory, które otrzymamy po tej transformacji będą nowym zbiorem danych dla klasyfikatora SVM. Autorzy pracy wykonali 10-fold cross validation w celu uzyskania bardziej wiarygodnych wyników.

3 Eksperymenty

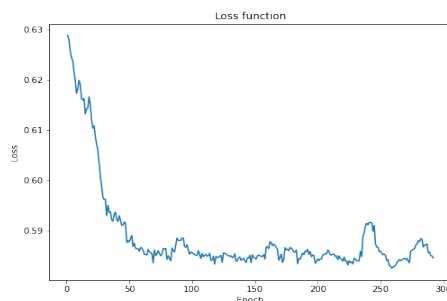
3.1 Wyniki

Przedstawimy teraz krzywe uczenia oraz dokładność klasyfikacji modeli GIN, GIN z warstwą uwagi i porównamy je z wynikami klasyfikacji dla baseline'u z pracy [CW18]. Wszystkie sieci były trenowane przez 300 epok, korzystając z optymalizatora Adama, gdzie stała ucząca wynosiła 0.01. Tak jak zostało to wspomniane, wyniki zostały osiągnięte na następujących zbiorach danych:

- PROTEINS



(a) Wartości funkcji straty dla zbioru danych PROTEINS dla GIN bez mechanizmu uwagi



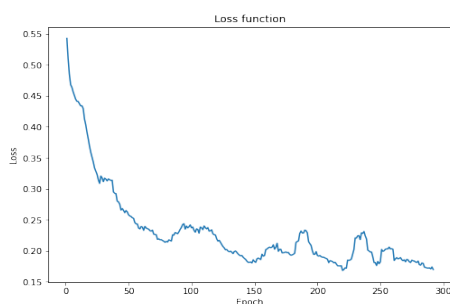
(b) Wartości funkcji straty dla zbioru danych PROTEINS dla GIN z mechanizmem uwagi

Widzimy, że w przypadku mechanizmu uwagi krzywa uczenia wygląda o wiele stabilniej. Obie krzywe wydają się maleć do zera.

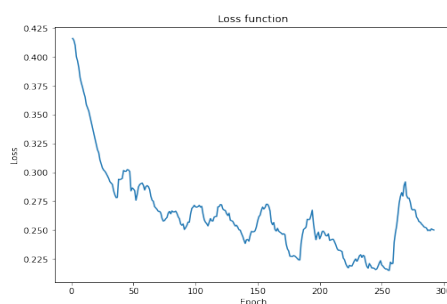
Dokładność klasyfikacji na zbiorze testowym:

- GIN bez uwagi: 74.22,
- GIN z uwagą: 74.74,
- LDP: 72.7.

• MUTAG



(a) Wartości funkcji straty dla zbioru danych MUTAG dla GIN bez mechanizmu uwagi



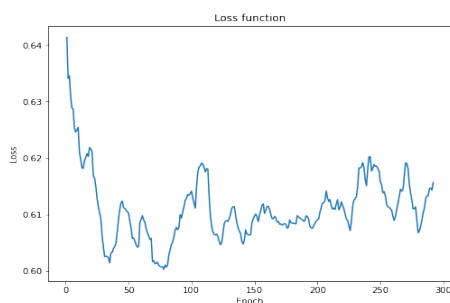
(b) Wartości funkcji straty dla zbioru danych MUTAG dla GIN z mechanizmem uwagi

Tym razem nieco stabilniej wygląda krzywa uczenia dla GIN bez mechanizmu uwagi. Obie krzywe wydają się maleć do zera.

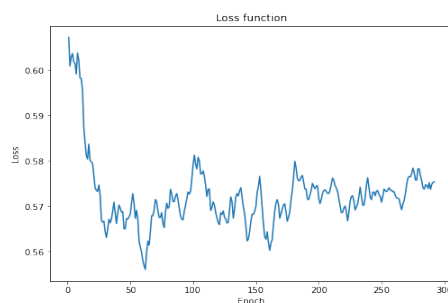
Dokładność klasyfikacji na zbiorze testowym:

- GIN bez atencji: 94.74,
- GIN z atencją: 78.95,
- LDP: 90.1.

• PTC_MM



(a) Wartości funkcji straty dla zbioru danych PTC_MM dla GIN bez mechanizmu atencji



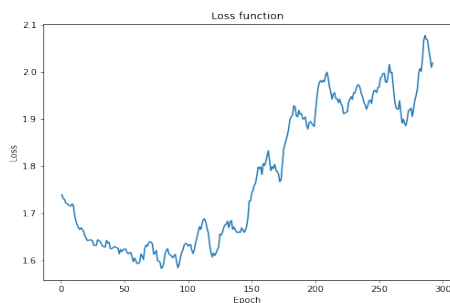
(b) Wartości funkcji straty dla zbioru danych PTC_MM dla GIN z mechanizmem atencji

Obie krzywe uczenia nie wyglądają za dobrze, ponieważ wydają się nie zbiegać do zera.

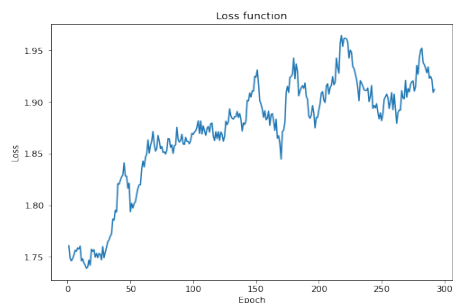
Dokładność klasyfikacji na zbiorze testowym:

- GIN bez atencji: 70.59,
- GIN z atencją: 73.53,
- LDP: 61.7 - tutaj warto wziąć poprawkę, że nie wiadomo, jaki dokładnie zbiór PTC autorzy benchmarku mieli na myśli. Na stronie TUDaset znalazłem ich aż 4 rodzaje.

• ENZYMES



(a) Wartości funkcji straty dla zbioru danych ENZYMES dla GIN bez mechanizmu uwagi



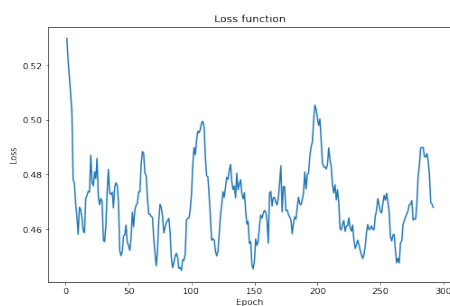
(b) Wartości funkcji straty dla zbioru danych ENZYMES dla GIN z mechanizmem uwagi

Obie krzywe uczenia wyglądają źle, ponieważ ich wartości wydają się dążyć do nieskończoności.

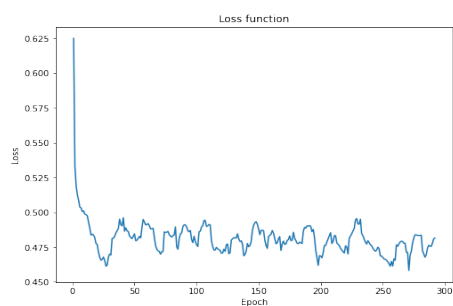
Dokładność klasyfikacji na zbiorze testowym:

- GIN bez uwagi: 41.67,
- GIN z uwagą: 33.33,
- LDP: 35.3.

• DD



(a) Wartości funkcji straty dla zbioru danych DD dla GIN bez mechanizmu uwagi



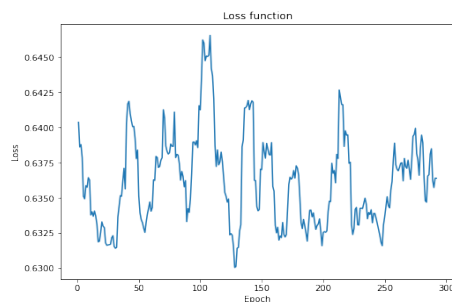
(b) Wartości funkcji straty dla zbioru danych DD dla GIN z mechanizmem uwagi

Tym razem krzywa uczenia dla GIN z mechanizmem uwagi wygląda dużo stabilniej i zbiega do zera.

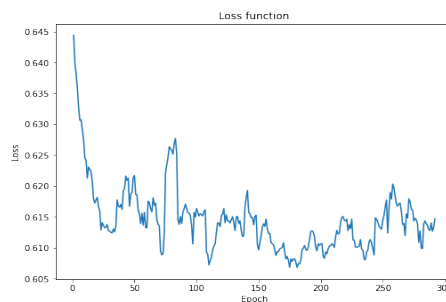
Dokładność klasyfikacji na zbiorze testowym:

- GIN bez uwagi: 80.3,
- GIN z uwagą: 85.56,
- LDP: 75.5.

- NCI1



(a) Wartości funkcji straty dla zbioru danych NCI1 dla GIN bez mechanizmu uwagi



(b) Wartości funkcji straty dla zbioru danych NCI1 dla GIN z mechanizmem uwagi

Obie krzywe uczenia nie wyglądają stabilnie.

Dokładność klasyfikacji na zbiorze testowym:

- GIN bez uwagi: 62.2,
- GIN z uwagą: 60.19,
- LDP: 73.

3.2 Wnioski

Przedstawimy teraz następujące wnioski wyciągnięte z powyższych rozważań:

- W większości przypadków architektura sieci neuronowej GIN i GIN z warstwą uwagi były w stanie przewyższyć pod względem dokładności klasyfikacji wynik uzyskany przez LDP.

- Im większa liczba grafów w zbiorze danych, tym trudniej było nauczyć modele czegoś sensownego.
- W przypadku klasyfikacji niebinarnej być może dobrym pomysłem byłby lepszy dobór parametrów do modelu oraz zwiększenie liczby epok, ponieważ uśrednione wartości funkcji straty zdają się uciekać w nieskończoność. Nie jesteśmy jednak teraz w stanie ocenić, czy jest to jedynie trend lokalny, czy globalny.
- Dla większej części zbiorów mechanizm uwagi wygładził fluktuacje wokół trendu dla funkcji strat.

4 Zawartość załącznika

W załączniku zamieszczamy:

- Dwa notatniki Pythona z kodem zawierającym zdefiniowane architektury sieci neuronowych oraz metodami do ich treningu oraz ewaluacji.
- Wykresy funkcji straty.
- Pliki tekstowe zawierające historię uczenia.
- Arkusz Excela z wartościami dokładności klasyfikacji.
- Raport w formacie PDF.

5 Bibliografia

References

- [CW18] Chen Cai and Yusu Wang. *A simple yet effective baseline for non-attributed graph classification*. 2018. DOI: 10.48550/ARXIV.1811.03508. URL: <https://arxiv.org/abs/1811.03508>.
- [Xu+18] Keyulu Xu et al. *How Powerful are Graph Neural Networks?* 2018. DOI: 10.48550/ARXIV.1810.00826. URL: <https://arxiv.org/abs/1810.00826>.

- [HV21] Ningyuan Teresa Huang and Soledad Villar. “A Short Tutorial on The Weisfeiler-Lehman Test And Its Variants”. In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, June 2021. DOI: 10.1109/icassp39728.2021.9413523. URL: <https://doi.org/10.1109/2Ficassp39728.2021.9413523>.