



AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Grafowe sieci neuronowe w zastosowaniach telekomunikacyjnych

Piotr Chołda, Krzysztof Rusek

Instytut Telekomunikacji AGH

Sekcja Telekomunikacji, Komitet EiT PAN, 26.05.2022

Plan prezentacji

- 1 **Potrzeba grafowych sieci neuronowych GNN**
- 2 **Przykład: GNN oparte na przekazie komunikatów MPNN**
- 3 **Przykład: przewidywanie opóźnień**

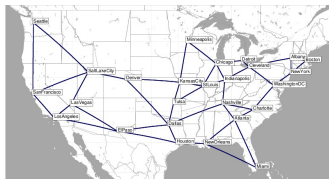
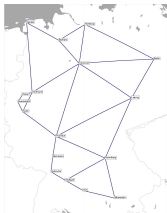
- 1** Potrzeba grafowych sieci neuronowych GNN
- 2 Przykład: GNN oparte na przekazie komunikatów MPNN
- 3 Przykład: przewidywanie opóźnień

Reprezentacja danych

Jak reprezentować dane grafowe na potrzeby uczenia maszynowego?

Sztuczne sieci neuronowe SSN: tensor WE — tensor WY

x — Warstwa SSN — y



Reprezentacja danych

Jak reprezentować dane grafowe na potrzeby uczenia maszynowego?

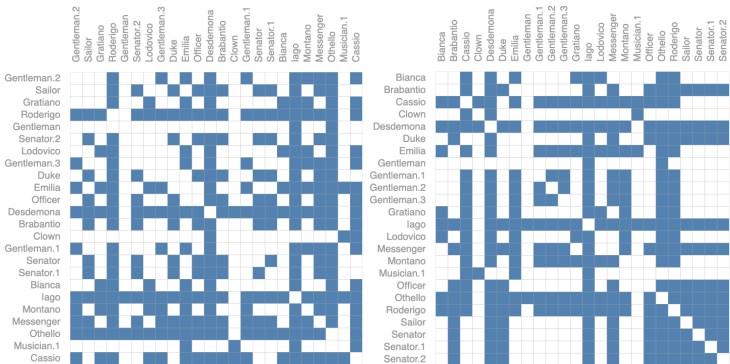
x — Warstwa SSN — y

- Sieci telekomunikacyjne: kratowe i **brak regularności** (+ różne liczby wierzchołków/krawędzi); wyróżniono pewne właściwości ogólne (np. rozkład potęgowy stopnia wierzchołka) — ale o charakterze probabilistycznym.
- Reprezentacja macierzowa (= tensor) nie jest rozwiązaniem:
 - sieci mają **różne liczby** wierzchołków,
 - bardzo **rzadka reprezentacja** (duża liczba zer w macierzy sąsiedztwa) na wejściu SSN jest problemem w uczeniu,
 - te same grafy mogą być różnie reprezentowane ze względu na **izomorfizm** (permutacja macierzy, a nie przesunięcie).

Reprezentacja danych

Jak reprezentować dane grafowe na potrzeby uczenia maszynowego?

Reprezentacja wspólnego występowania postaci w scenach *Otella*:



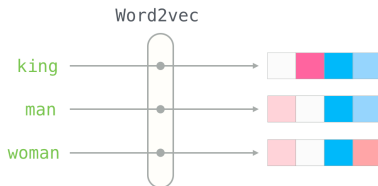
Macierz sąsiedztwa nie może być przetwarzana jako obraz (np. z użyciem sieci spłotowych).

Reprezentacja danych

Zanurzenia (*embeddings*)

- Rozwiązanie: reprezentacja struktury jako **wektora liczb rzeczywistych**.
- Zanurzenie ma charakter **homomorficzny**: zachowuje relacje między obiektami (tj. podobne grafy są reprezentowane przez podobne wektory).
- Zanurzenia:
 - Najpopularniejsze — zanurzenia słów w przetwarzaniu języka naturalnego NLP. Posłużymy się nimi do pokazania intuicji.
 - Typowe zanurzenie jest wstępnym etapem przetwarzania danych, stanowi dopiero pierwszy etap właściwej metody klasyfikacyjnej, regresyjnej itp.
 - Przykład zastosowania **uczenia nienadzorowanego**. Zaleta: brak potrzeby etykietowania itp. (*annotation*) — wystarczy po prostu skorzystać ze znanej topologii sieci (np. reprezentowanej przez graf ważony).
 - Można też łączyć z uczeniem nadzorowanym.

Przykład pożądanego zanurzenie dla słów:



- Typowo: **brak interpretowalności**.
- Reprezentacje wektorowe są atrakcyjne — mimo że reprezentują dane kategoryczne możemy na nich wykonywać obliczenia:

$$f(\text{queen}) \approx f(\text{king}) - f(\text{man}) + f(\text{woman})$$

Plan prezentacji

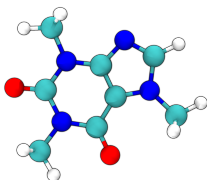
- 1 Potrzeba grafowych sieci neuronowych GNN
- 2 Przykład: GNN oparte na przekazie komunikatów MPNN**
- 3 Przykład: przewidywanie opóźnień

- GNN: reprezentacja grafu uzyskiwana z użyciem sieci neuronowych.
- Pojedyncza sieć telekomunikacyjna (którą utożsamiamy z grafem ważonym) jest reprezentowana z użyciem wektora o arbitralnie dobranej wymiarowości:

$$\text{GNN} : \mathcal{G} \longrightarrow \mathbf{h} \in \mathbb{R}^n$$

- Graf jest reprezentowany z użyciem tzw. stanu ukrytego (*hidden state*): podobna koncepcja jest używana w przypadku przetwarzania kontekstowego w sieciach rekurencyjnych.
- Opisywane podejście używa tzw. **przekazywania komunikatów w oparciu o sieci neuronowe** MPNN (*message-passing neural networks*).

Badania w naukach chemicznych — przewidywanie właściwości różnych związków z użyciem uczenia maszynowego:



Podstawowa koncepcja

Reprezentacja grafu za pośrednictwem „wiedzy” poszczególnych węzłów — każdy z nich ma własny stan ukryty reprezentowany w postaci (nieinterpretowalnego) wektora.

Przekaz komunikatów:

- abstrakcyjna operacja, nie ma **nic wspólnego** z przekazywaniem danych w sieci telekomunikacyjnej;
 - służy do modelowania wiedzy lokalnej (poszczególnych wierzchołków) nt. całego grafu (wiedzy globalnej).
-
- Wierzchołek $v \in \mathcal{V}$ jest charakteryzowany wektorem cech \mathbf{x}_v (typ wierzchołka, wielkość ruchu wchodzącego itp.).
 - Łuk $(v, w) \in \mathcal{E}$ jest charakteryzowany wektorem cech $\ell_{(v,w)}$ (np. przepływność bitowa).
 - Na podstawie struktury grafu $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ oraz cech wierzchołków i łuków otrzymamy stany ukryte dla każdego z wierzchołków \mathbf{h}_v .

Stan ukryty wierzchołka

- Poszukujemy stanu wierzchołka \mathbf{h}_v (dla każdego $v \in \mathcal{V}$).
- Operacja ma charakter iteracyjny w oparciu o funkcję opisywaną jako F :

$$\mathbf{h}_v^{(i+1)} = \sum_{(w,v) \in \mathcal{V}} f(\mathbf{x}_v, \mathbf{x}_w, \ell_{(v,w)}, \mathbf{h}_w^{(i)})$$

- Wymiar wektora reprezentującego stan wierzchołka \mathbf{h}_v nie zależy od struktury sieci (ale jego wartość już tak).
- Funkcja f jest obliczana z użyciem przekazu komunikatów.

Niezależnie od wartości startowej $\mathbf{h}^{(0)}$

$$\mathbf{h}^{(0)}, \mathbf{h}^{(1)} = F(\mathbf{h}^{(0)}), \dots, \mathbf{h}^{(i+1)} = F(\mathbf{h}^{(i)}), \dots, \mathbf{h}^{(m+1)} = \mathbf{h}^{(m)} = F(\mathbf{h}^{(m)})$$

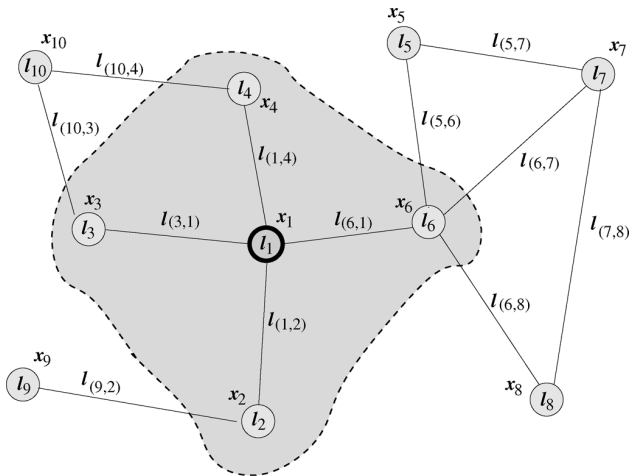
Jeśli F jest odwzorowaniem zwężającym, to niezależnie od punktu startowego znajdziemy **unikalny punkt stały**:

$$\mathbf{h} = F(\mathbf{h})$$

W praktyce zbieżność jest osiągnięta bardzo szybko: 5–7 iteracji.

GNN

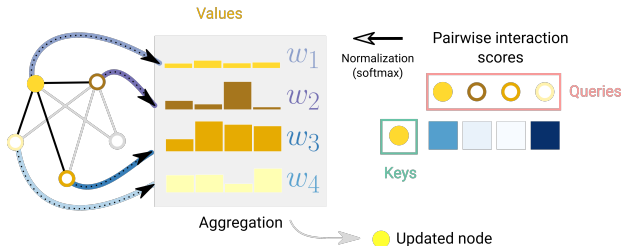
Sposób wyliczania F na podstawie przekazu komunikatów



Komunikat (*message*)

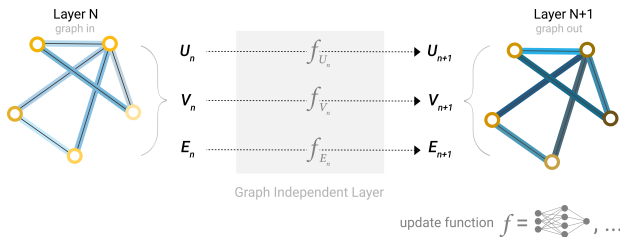
$$\mathbf{m}_v^{(i+1)} = \sum_{w:(v,w) \in \mathcal{E}} M(\mathbf{h}_v^{(i)}, \mathbf{h}_w^{(i)}, \ell_{(v,w)}) \quad M : \text{sieć neuronowa}$$

Stosujemy mechanizm uwagi (jak w przypadku transformerów, które służą do przetwarzania języka naturalnego):



Aktualizacja (*update*)

$$\mathbf{h}_v^{(i+1)} = U \left(\mathbf{h}_v^{(i)}, \mathbf{m}_v^{(i+1)} \right) \quad (\text{np.: } \mathbf{h}_v^{(0)} = \mathbf{x}_v) \quad U : \text{sieć neuronowa}$$

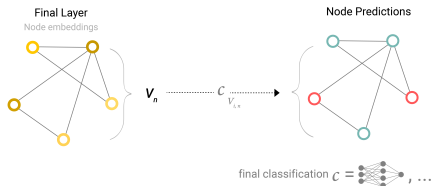


Reprezentacja grafu — zanurzenie

$\mathbf{h} = \sum_{v \in \mathcal{V}} f(\mathbf{h}_v)$ reprezentacja grafu na podstawie sumy stanów

Odczyt na podstawie zanurzenia grafu

$\hat{y} = R(\mathbf{h})$ R : sieć neuronowa



Plan prezentacji

- 1 Potrzeba grafowych sieci neuronowych GNN
- 2 Przykład: GNN oparte na przekazie komunikatów MPNN
- 3 Przykład: przewidywanie opóźnień**

- Przykład wygodny do uczenia: model analityczny na potrzeby uczenia nadzorowanego.
- Λ_v : natężenie ruchu w wierzchołku v .
- μ_v : intensywność obsługi w wierzchołku v .
- Trasowanie ruchu na podstawie macierzy ratingu $[r_{vw}]$.
- Uczenie: sieć losowa (według modelu Barabasi'ego–Albert).

Model analityczny średniego opóźnienia w sieci

$$\lambda_v = \Lambda_v + \sum_w r_{vw} \lambda_w \quad v \in \mathcal{V}$$

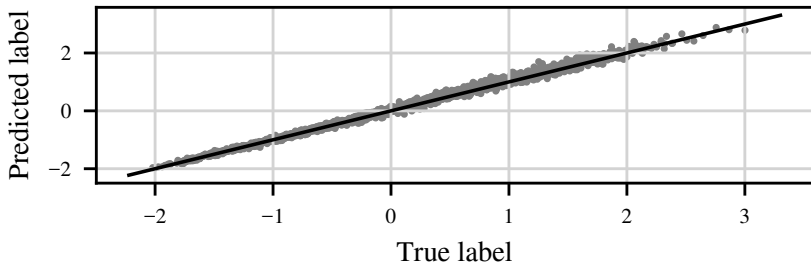
$$W = \frac{\sum_v q_v}{\sum_v \Lambda_v} \quad q_v = \frac{\lambda_v}{\mu_v - \lambda_v} \quad \text{na podstawie prawa Little'a}$$

Konstrukcja GNN na podstawie MPNN:

- Cechy wierzchołka: natężenie ruchu oraz intensywność obsługi
 $\mathbf{x}_v = [\lambda_v, \mu_v]$.
- Cechy łuków są oparte jedynie na macierzy rutowania:
 $\ell_{(v,w)} = [r_{vw}]$.
- Odczyt wzorcowy y : poszukiwane opóźnienie średnie w sieci W .

Konstrukcja GNN na podstawie MPNN (cd.):

- Uczenie: losowo generujemy sieci, dla nich losujemy wartości Λ , μ , r , analitycznie wyliczamy W i stosujemy proces przekazywania komunikatów w celu wyliczenia \mathbf{h} (w oparciu o sieci M i U).
- Na podstawie stanu wewnętrznego dokonujemy odczytu $\hat{y} = R(\mathbf{h})$, który jest porównywany z odczytem wzorcowym W .
- Na podstawie różnicy między \hat{y} i W następuje z użyciem algorytmu wstecznej propagacji aktualizacja parametrów wszystkich składowych sztucznych sieci neuronowych używanych do wyliczenia zanurzenia (M, U, R) .



Rysunek: Wyniki dla prognozowania W

Zestaw uczący: 50.000 losowych sieci BA. SSN: w oparciu o MLP i RNN. Zestaw testowy: 1000 losowych konfiguracji opartych na istniejących sieciach (SNDlib).

Podsumowanie

- Potrzeba reprezentacji dowolnych struktur sieciowych (grafów ważonych) w celu przetwarzania z użyciem narzędzi uczenia maszynowego.
- Zanurzenia grafowe: uniwersalna reprezentacja z użyciem wielowymiarowych wektorów rzeczywistych.
- Wykorzystanie wielu pomocniczych sieci neuronowych w celu uzyskania zanurzenia (wymiana komunikatów, aktualizacja, odczyt).

**Bardzo Państwu dziękuję
za uwagę**