Uvod v globoko učenje in grafovske nevronske mreže

Ljupčo Todorovski

Univerza v Ljubljani, Fakulteta za matematiko in fiziko
Institut Jožef Stefan, Odsek za tehnologije znanja (E8)

Maj 2023

Pregled predavanja

1. Uvod v globoko učenje
   - Osnovne definicije
   - Običajne funkcije aktivacije
   - Vzvratno razširjanje napake, *back propagation*
   - Delovanje nevronskih mrež
2. Grafovske nevronske mreže
   - Od grafa do strukture GNM
   - Struktura in učenje GNM

Nevroni in sinapse

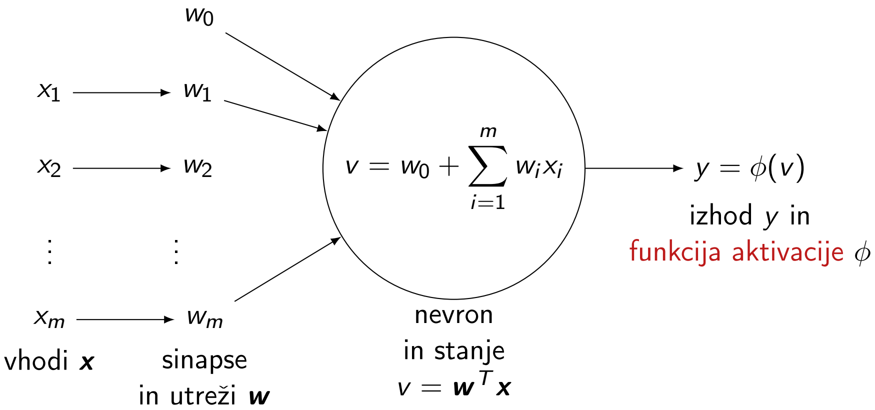
Gradnika nevronske mreže (NM)

- Nevron: ima stanje $v \in \mathbb{R}$ in izhod $y \in \mathbb{R}$
- Sinapsa: povezava med nevroni, ki ima utež $w \in \mathbb{R}$
- Sinapse določajo strukturo (tudi arhitekturo, topologijo) NM

Izvajanje nevronske mreže

- Vhodni podatki spremenijo stanje in izhode izbranih, vhodnih nevronov
- Spremenjeni izhodi nevronov se prenašajo po sinapsah
- Nevron, ob spremembi vhodov na sinapsah, izračuna (spremeni) izhod
- Izračun se nadaljuje, dokler se izhodi nevronov ne ustalijo

Funkcija nevrona in funkcija aktivacije ϕ



Vektorska notacija: $\boldsymbol{x} = (1, x_1, x_2, \ldots x_m)^T$ in $\boldsymbol{w} = (w_0, w_1, \ldots w_m)^T$.

Struktura usmerjenih nevronskih mrež

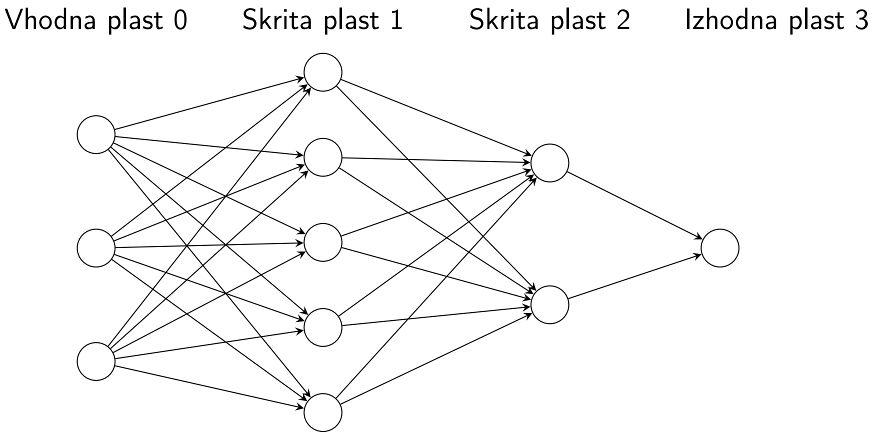
Nevroni urejeni v plasteh z indeksi od 0 do $L + 1$

- Vhodna plast 0: en nevron za vsako vhodno (numerično) spremenljivko
- Izhodna plast $L + 1$
  - Regresija: en izhodni nevron
  - Klasifikacija: en izhodni nevron za vsako vrednost ciljne spremenljivke
- $L \geq 0$ skritih, *hidden* plasti

Sinapse in uteži

- Med nevroni v eni plasti ni povezovalnih sinaps
- Sosednji plasti $l - 1$ in l za $l = 0 \ldots L$ polno povezani: sinapse povezujejo vsak nevron plasti $l - 1$ z vsakim nevronom plasti l

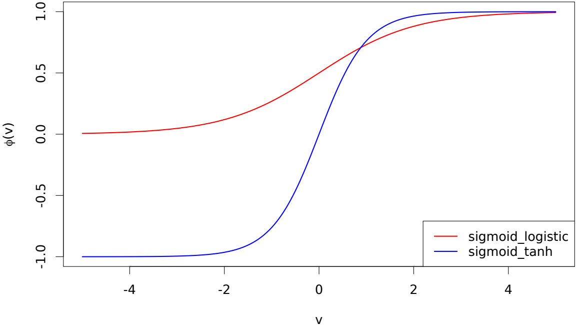
Primer usmerjene nevronske mreže: Dve skriti plasti



Koliko uteži ima ta usmerjena nevronska mreža?

Sigmoidni funkciji aktivacije v usmerjenih NM

- Logistična $\phi(v) = \frac{1}{1+e^{-v}}$, za odvod velja $\phi'(v) = \phi(v)(1 - \phi(v))$
- Hiperbolična $\phi(v) = \tanh v$, za odvod velja $\phi'(v) = 1 - \phi(v)^2$



Funkcija aktivacije za izhodno plast klasifikacijske NM

$$\phi_{softmax}(v) = \frac{e^v}{\sum_k e^{v_k^{(L+1)}}}$$

- Indeks k gre skozi vse nevrone izhodne plasti $L+1$
- Zagotovi, da je seštevek izhodov vseh nevronov izhodne plasti 1
- Zato lahko izhode interpretiramo kot verjetnosti, torej
- Za $D_Y = \{v_1, v_2, \ldots v_c\}$ je izhod $y_k = P(Y = v_k)$, $k = 1 \ldots c$

Notacija v izpeljavah in formulah

Indeksi plasti nevronov

- 0: vhodna plast, plast vhodnih nevronov
- $L+1$: izhodna plast, plast izhodnih nevronov
- $1, 2, \ldots, L$: skrite plasti, plasti skritih nevronov

Uteži sinaps ter stanja in izhodi nevronov

- $w_{ji}^{(l)}$ je utež sinapse med j-tim nevronom plasti $l-1$ in i-tim nevronom plasti l
- $v_i^{(l)}$ je stanje i-tega nevrona plasti l, velja torej $v_i^{(l)} = \sum_j w_{ji}^{(l)} y_j^{(l-1)}$
- $y_i^{(l)}$ je izhod i-tega nevrona plasti l, $y_i^{(l)} = \phi\left(v_i^{(l)}\right)$

Funkcija izgube za regresijo in njen odvod

Kvadratna napaka

$$E = \frac{1}{2}\sum_i (y_i - \hat{y}_i)^2 = \frac{1}{2}\sum_i \left(y_i - y_i^{(L+1)}\right)^2,$$

ker velja $\hat{y}_i = y_i^{(L+1)}$.

Odvod funkcije izgube

$$\frac{\partial E}{\partial \hat{y}_i} = -(y_i - \hat{y}_i) = -\left(y_i - y_i^{(L+1)}\right)$$

Funkcija izgube za klasifikacijo

Prečna entropija H med

- Opazovano porazdelitvijo p vrednosti ciljne spremenljivke Y
- Porazdelitvijo q vrednosti Y, ki jo napove NM, $q(v_i) = y_i^{(L+1)}$

$$E = H(p, q) = -\sum_{v_i \in D_Y} p(v_i) \log_2 q(v_i) = -\sum_{v_i \in D_Y} p(v_i) \log_2 y_i^{(L+1)}$$

Za podan učni primer z vrednostjo ciljne spremenljivke $Y = v_m$

$$E = -\log_2 q(v_m) = -\log_2 y_m^{(L+1)},$$

ker velja $p(v_m) = 1, \forall v_i \in D_Y, v_i \neq v_m : p(v_i) = 0$.

Odvod prečne entropije

$$\frac{\partial E}{\partial \hat{y}_i} = \begin{cases} 0 & ;\, i \neq m \\ -\frac{1}{y_m^{(L+1)} \log 2} & ;\, i = m \end{cases}$$

Gradienti izgube E za izhodno plast

$$\hat{y}_i = y_i^{(L+1)} = \phi\left(v_i^{(L+1)}\right),\ v_i^{(L+1)} = \sum_j w_{ji}^{(L+1)} y_j^{(L)}$$

$$\begin{aligned}
\frac{\partial E}{\partial w_{ji}^{(L+1)}} &= \frac{\partial E}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial v_i^{(L+1)}} \frac{\partial v_i^{(L+1)}}{\partial w_{ji}^{(L+1)}} \\
&= -\left(y_i - y_i^{(L+1)}\right) \phi'\left(v_i^{(L+1)}\right) \frac{\partial \sum_k w_{ki}^{(L+1)} y_k^{(L)}}{\partial w_{ji}^{(L+1)}} \\
&= -\left(y_i - y_i^{(L+1)}\right) \phi'\left(v_i^{(L+1)}\right) y_j^{(L)}
\end{aligned}$$

$$\Delta w_{ji}^{(L+1)} = \eta \left(y_i - y_i^{(L+1)}\right) \phi'\left(v_i^{(L+1)}\right) y_j^{(L)}$$

Gradienti izgube za uteži skritih plasti, posredno odvajanje

$$\frac{\partial E}{\partial w_{ji}^{(l)}} = \frac{\partial E}{\partial y_i^{(l)}} \frac{\partial y_i^{(l)}}{\partial v_i^{(l)}} \frac{\partial v_i^{(l)}}{\partial w_{ji}^{(l)}} = \frac{\partial E}{\partial y_i^{(l)}} \phi'(v_i^{(l)})\, y_j^{(l-1)}$$

Trik za izračun $\partial E/\partial y_i^{(l)}$ iz $\partial E/\partial y_k^{(l+1)}$

$$\begin{aligned}
\frac{\partial E}{\partial y_i^{(l)}} &= \sum_k \frac{\partial E}{\partial v_k^{(l+1)}} \frac{\partial v_k^{(l+1)}}{\partial y_i^{(l)}} \\
&= \sum_k \frac{\partial E}{\partial y_k^{(l+1)}} \frac{\partial y_k^{(l+1)}}{\partial v_k^{(l+1)}} \frac{\partial v_k^{(l+1)}}{\partial y_i^{(l)}} \\
&= \sum_k \frac{\partial E}{\partial y_k^{(l+1)}} \phi'\left(v_k^{(l+1)}\right) w_{ik}^{(l+1)}
\end{aligned}$$

Gradienti izgube za uteži skritih plasti

$$\Delta w_{ji}^{(l)} = \eta \, \phi' \left(v_i^{(l)} \right) \, y_j^{(l-1)} \sum_k \frac{\partial E}{\partial y_k^{(l+1)}} \, \phi' \left(v_k^{(l+1)} \right) \, w_{ik}^{(l+1)}$$

- Iteracija pravila od skrite plasti L do skrite plasti 1
- V vhodni plasti 0 upoštevamo $y_i^{(0)} = x_i$, t.j., vrednost spremenljivke X_i

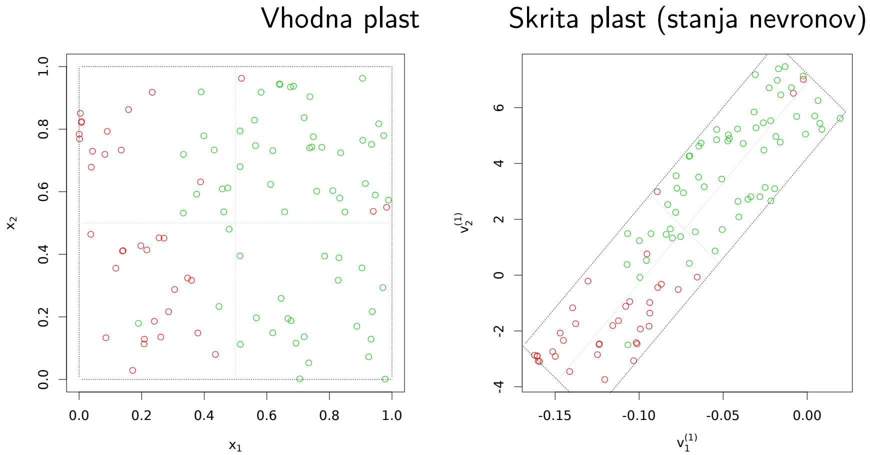
Stohastična različica učnega algoritma

Osnovna ideja: vzorčenje primerov

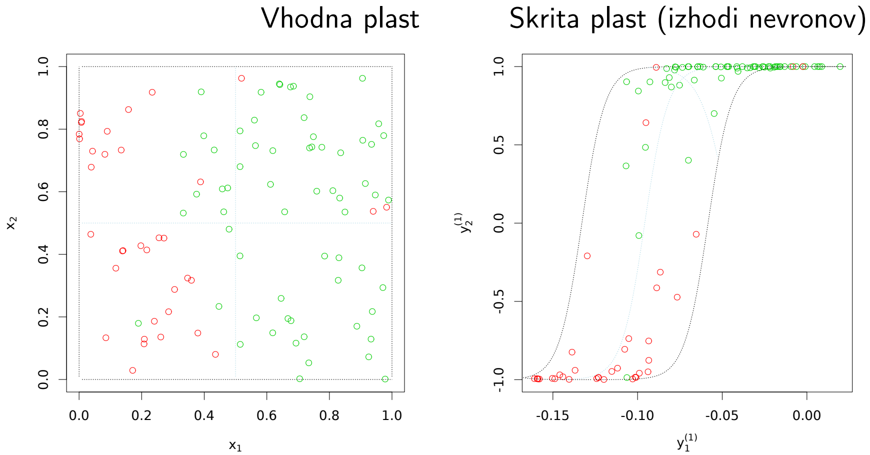
- Učenje v šaržah oz. *batch*-ih primerov, velikost šarže je nadparameter
- V vsaki iteraciji izberemo šaržo za vzvratno razširjanje napake
- Z matričnimi verzijami formul lahko gradiente računamo zelo učinkovito, za celo šaržo primerov naenkrat

- Število iteracij je število primerov deljeno z velikostjo šarže
- Epoha je en sprehod skozi celotno podatkovno množico

Plast skritih nevronov in nelinearnost pri klasifikaciji



Plast skritih nevronov in nelinearnost pri klasifikaciji



Določanje strukture

Zaporedje plasti različnih tipov

- Običajna plast: polno povezana s prejšnjo
- Redka plast: polno povezana z regularizacijo uteži
- Običajni in redki plasti lahko dodelimo poljubno funkcijo aktivacije

- Konvolucijska plast: za obdelavo slik, implementira konvolucijski filter
- Akumulacijska plast: združi izhode nevronov iz prejšnje plasti

Konvolucijska, *convolution*, plast

Notacija

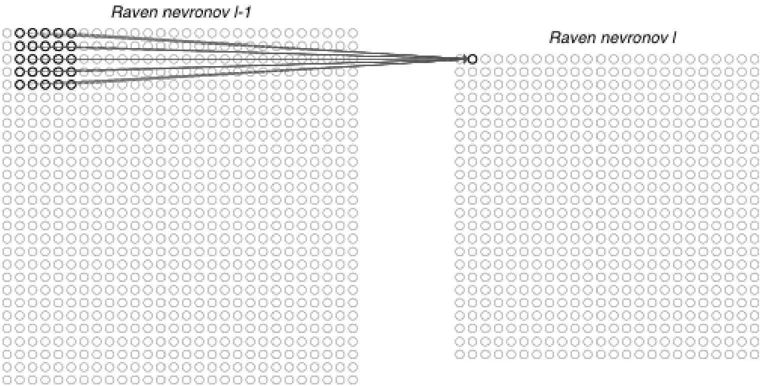
- Nevroni v plasti $l-1$ organizirani v matriko dimenzij $x \times y$
- Uteži w_0 in konvolucijska matrika dimenzij $c \times c$, elementi $w_{i,j}$
- Plast l je matrika nevronov dimenzij $(x-c+1) \times (y-c+1)$

Konvolucija, uporabljena namesto običajne linearne obtežene vsote

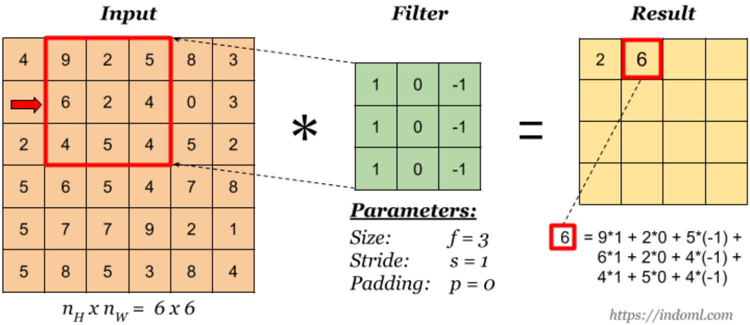
$$v_{i,j}^{(l)} = w_0 + \sum_{k=1}^{c} \sum_{m=1}^{c} w_{k,m} y_{i+k-1,j+m-1}^{(l-1)}$$

- $y_{i,j}^{(l-1)}$ in $v_{i,j}^{(l)}$ so izhodi in stanja nevronov v plasti $l-1$ oz. l
- Običajna dimenzija konvolucijske matrike (tudi filtra) je $c = 5$

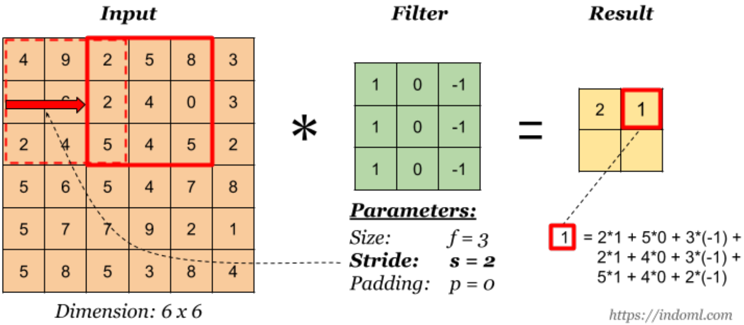
Konvolucijska plast: Grafični prikaz



Konvolucijska plast: Primer izračuna



Konvolucijska plast: Dodaten parameter korak, *stride*



Akumulacijska, *pooling*, plast

Notacija

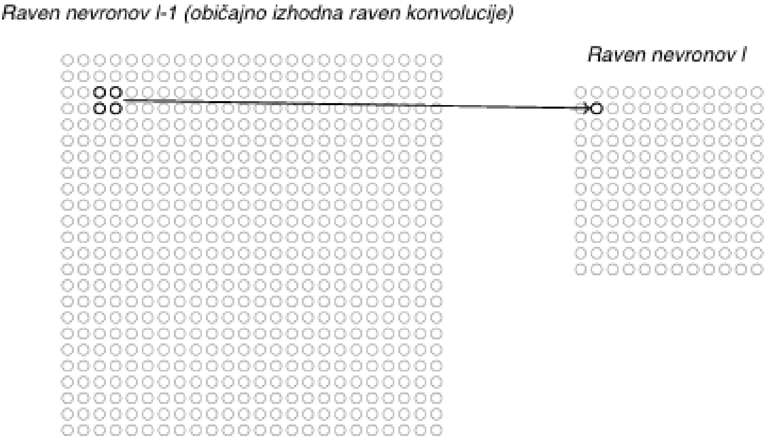
- Nevroni v plasti $l-1$ organizirani v matriko dimenzij $x \times y$
- Plast l je matrika nevronov dimenzij $(x/a) \times (y/a)$
- Predpostavka: $a|x$ in $a|y$

max-akumulacija, uporabljena namesto običajne linearne obtežene vsote

$$v_{i,j}^{(l)} = \max \left\{ y_{(i-1)\cdot a+1,(j-1)\cdot a+1}^{(l-1)}, \ldots, y_{i\cdot a,j\cdot a}^{(l-1)}) \right\}$$

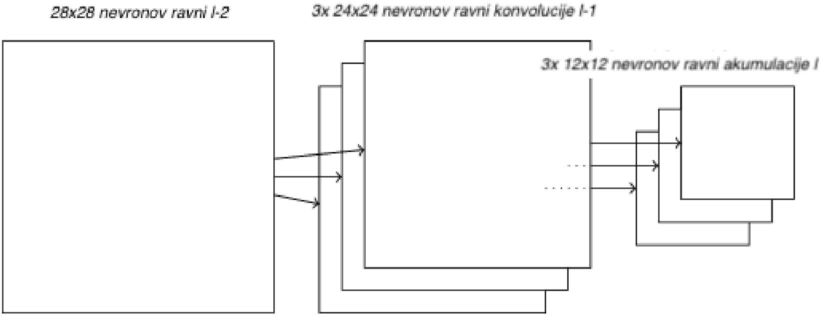
- $y_{i,j}^{(l-1)}$ in $v_{i,j}^{(l)}$ so izhodi in stanja nevronov v plasti $l-1$ oz. l
- Običajna vrednost $a=2$

Raven akumulacije: Grafični prikaz



Običajna topologija: $r\times$konvolucija $\to$ $r\times$akumulacija

Uspešna uporaba za razpoznavanje z roko napisnih številk.



Nevronske mreže in grafi izračunov

Nevronska mreža določa graf izračuna funkcije

- Preslika vrednosti vhodnih spremenljivk X_i, $i = 1 \ldots p$
- V vrednost izhodne/ izhodnih spremenljivk(e) Y

Struktura grafa izračuna GNM

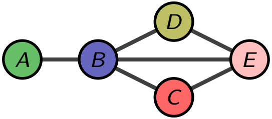
Funkcija, ki

- Za podana vpetja vozlišč iz grafovske soseščine vozlišča v
- Izračuna vpetje v na izhodu

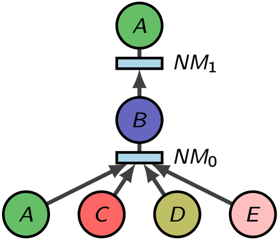
Bolj splošen pristop od običajnih vpetij, npr. node2vec

- Lahko izračuna vpetja za nova vozlišča, ki jih ni bilo v času učenja
- Lahko dobljena vpetja vgradimo v večjo nevronsko mrežo
- Slednja lahko poskrbi za poljubno nalogo klasifikacije ali regresije

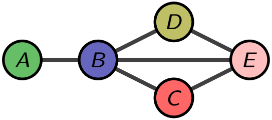
Od navadnega do grafa izračuna za vozlišče A



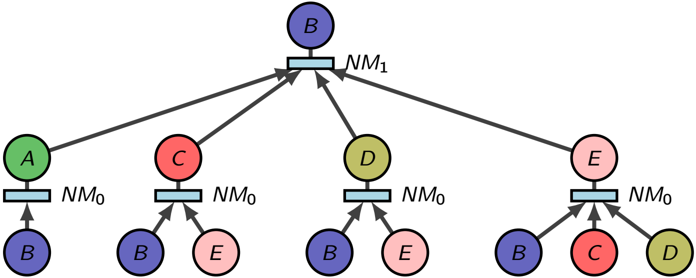
Graf izračuna globnine 2



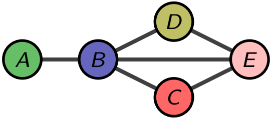
Od navadnega do grafa izračuna za vozlišče B



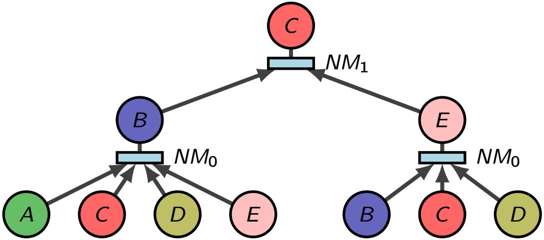
Računski graf globnine 2



Od navadnega do grafa izračuna za vozlišče C



Računski graf globnine 2



Nevronske mreže NM_I

Enotna struktura

- Izhod: vpetje opazovanega vozlišča
- Vhodi: vpetja sosednjih vozlišč v grafu

Težava in rešitev

- Težava: Spreminjajoče se število vhodov (sosednjih vozlišč)
- Rešitev: Akumulacijska plast združi vhode v vektor fiksne dolžine
- Združevanje mora biti simetrično, vrstni red argumentov brez vpliva
- Primer take funkcije združevanja je povprečje

Nevronske mreže NM_l in njihovo učenje

Večplastne nevronske mreže

- Akumulacijska plast, ki poskrbi za združevanje vhodov
- Ostale plasti so lahko poljubne, običajno kar polno povezane plasti

Različice osnovne strukture

- Grafovske konvolucijske mreže, GCN
- GraphSAGE

Splošna struktura NM_l: Notacija

$$h^{(0)} = x_v$$
$$h_v^{(l+1)} = \phi \left(W_l \sum_{u \in \mathcal{N}(v)} \frac{h_u^{(l)}}{|\mathcal{N}(v)|} + B_l \, h_v^{(l)} \right)$$

- $h_v^{(l)}$ je izhodni vektor vpetja vozlišča v v plasti $l-1$
- $l = 1, \ldots, L-1$, $h_v^{(L)}$ je končno vpetje
- x_v je vektor vrednosti vhodnih spremenljivk v vozlišču v
- ϕ je (nelinearna) funkcija aktivacije, aplicirana po komponentah
- W_l in B_l sta matriki parametrov nevronske mreže NM_l
- $\mathcal{N}(v)$ je množica sosedov vozlišča v v grafu

Splošna struktura NM_i: Pomen

$$h_v^{(l+1)} = \phi\left(W_l \sum_{u \in \mathcal{N}(v)} \frac{h_u^{(l)}}{|\mathcal{N}(v)|} + B_l\, h_v^{(l)}\right)$$

- Združevanje vhodov, t.j., vpetij sosednih vozlišč iz prejšnje plasti
- Vpetje vozlišča v iz prejšnje plasti
- ϕ poskrbi za nelinearno transformacijo vpetja

- Transformacija je očitno odvedljiva, ni težav z računanjem gradientov

Matrična verzija formule

$$H^{(l+1)} = \phi\left(D^{-1}AH^{(l)}\,W_l^T + H^{(l)}\,B_l^T\right)$$

- $H^{(l)}$ sta matriki vpetij učnih vozlišč grafa v plasteh $l-1$ in l
- D je diagonalna matrika s stopnjami učnih vozlišč grafa
- A je matrika sosednosti učnih vozlišč grafa
- Funkcijo aktivacije ϕ apliciramo po elementih matrike

- Faktor $D^{-1}A$ poskrbi za izračun povprečja vpetij

Osnovna različica: Grafovske konvolucijske mreže, GCN

$$H^{(l+1)} = \phi\left(D^{-1}AH^{(l)}\,W_L^T\right)$$

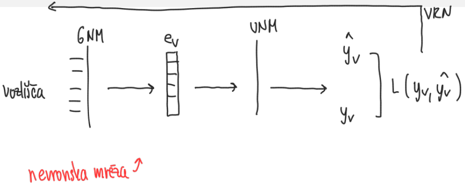
- Ne upošteva vložitve opazovanega vozlišča iz prejšnje plasti
- Funkcija združevanja je lahko zgolj navadno povprečje

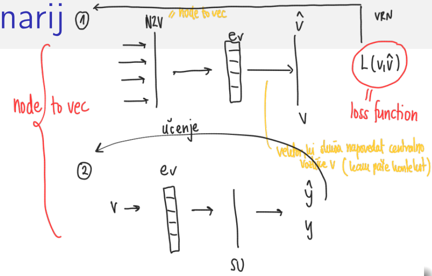
Različica GraphSAGE

$$H^{(l+1)} = \phi\left(concat\left(h_v^{(l)}, \operatorname*{agg}_{u \in \mathcal{N}(v)} \left(h_u^{(l)} \right) \right) W_l{}^T \right)$$

- Upošteva vložitev opazovanega vozlišča iz prejšnje plasti
- Funkcija združevanja je lahko poljubna, akumulacijska plast
- Običajno normaliziramo vpetja posameznih vozlišč na normo 1

Funkcija izgube: nadzorovani scenarij





Napovedovanje ciljne spremenljivke Y

- Primerjamo napoved $\hat{y}_v = f\left(h_v^{(L)}\right)$ in vrednost y_v v vozlišču v
- f je nevronska mreža: vhod je vpetje vozlišča, izhod pa napoved Y
- Če je ciljna spremenljivka numerična, uporabimo kvadratno napako
- Če je ciljna spremenljivka diskretna, uporabimo prečno entropijo

Funkcija izgube: nenadzorovani scenarij

Napovedovanje razdalje med vozlišči, podobno kot node2vec

- Učni primeri ustrezajo parom vozlišč u in v
- Vrednosti vhodnih spremenljivk dobimo s stikom vpetij u in v
- Ciljna spremenljivka je 1, če sta u in v podobna, sicer 0
- Podobnost vozlišč dobimo s pomočjo bolj enostavne metode za vpetje

Odločitve pri sestavljanju GCN

Odločitve glede strukture

- Število plasti, koliko plasti soseščin bomo uporabili
- Različica strukture GCN in funkcije združevanja
- Globina in struktura vsake plasti, NM_l

Funkcija izgube

- Nadzorovani ali nenadzorovani scenarij
- Opazovan objekt: vozlišče, povezava ali celoten graf

Težave pri sestavljanju GCN

Kako izbrati ustrezne globine?

- Če povečujemo število plasti, upoštevamo preveliko soseščino
- Če poglabljamo posamezne NM_l, tvegamo preprileganje

Kako izbrati ustrezne funkcije združevanja?

Oboje težavno, empirično primerjanje različnih možnosti za podatke.

Reference in implementacije

Uvod v grafovske nevronske mreže

- Začetek (Kipf in Welling 2017), `arXiv:1609.02907`
- Dober, ne preveč tehničen uvod, `distill.pub/2021/gnn-intro/`

Številne implementacije, priporočena knjižnica PyG

- Dostopna na `pytorch-geometric.readthedocs.io` ali `pyg.org`
- Vadnica v rubriki `Getting started/ Introduction by Example`