Benjamín Benčík
SFG April 27, 2023

Grafové neuronové site

Úvod

Konvolučné neuónové siete sú v dnešnej dobe už známe a často používané na obrazových dátach, kde dosahujú veľmi dobré vysledky. Preto sa neskôr začali aplikovať aj na viacej všeobecné grafové dáta. Obrázok si môžeme predstaviť ako homogénny graf kde každý vrchol reprezentuje pixel a hrany sú susedia daného pixelu. Všetobecné grafy môžu mať komplexnejšiu štruktúru a preto je ich aj o niečo náročnejšie spracovať pomocou konvolúcie. To však veľmi užitočnú reprezentáciu opmocou ktorej sa dajú modelovať problémy vo fyzike. V prípade tejto práce sme sa snažili modelovať systémy častic a následne naučiť neuronovú sieť odhadovať ich vlastnosti.

Popis úlohy

Cieľom práce bolo natrénovať sieť na odhadovanie vlastností systémov častic. Jednalo sa o regresné ulohy, kde mala sieť predikovať celkovú energiu systému. V projekte som pracoval s dvoma fyziklnými modelmi a to s Isingovým modelom a Heisenbergovým modelom.

1. Isingov model je model, ktorý sa používa na popisovanie ferromagnetických materiálov. V modeli je každá častica reprezentovaná ako spin, ktorý môže mať hodnoty $\{-1, 1\}$. V modeli je definovaná interakcia medzi časticami, ktorá je v grafe reprezentovaná ako váha hrany. V grafe je každý vrchol reprezentovaný ako spin a hrany reprezentujú interakciu medzi spinmi.

2. Heisenbergov model je funguje veľmi podobne ako Isingov model, ale v tomto prípade je každý spin reprezentovaný ako 3-dimenzionálny jednotkový vektor.

Pre oba modeli sme počítali hodnotu Hamiltoniánu: $\sum_{<i,j>} J_{i,j}\sigma_i\sigma_j$. Dôvod na použitie Isingovho modelu bol v tom, že je o niečo jednoduchší a chceli sme najskôr sieť vyskúšať na jednoduchej úlohe a až potom sme trénovali Heisenbergov model s viacero dimanziami. V prípade Heisenbergovho modelu sme chceli overiť, či sieť dokáže pracovať s viacero dimenziami.

Generovanie dát

Pre túto úlohu bolo najvhodnejšie dáta generovať, čo nám dávalo prakticky nebmedzenú trénovaciu množinu. Sieť som trénoval na dvoch typoch grafov. Jednoduchšia varianta bola ceta a tá kompikovanejšia periodická mriežka. Teda ak máme mriežku veľkosti $N \times N$ a množinu vrcholov V tak bude pridaná hrana medzi vrcholy $(V_{0,0}, V_{0,n}), (V_{0,0}, V_{n,0}), (V_{0,0}, V_{n,n})$ a tak ďalej. Všetky hodnoty váh boli generované z normálneho rozdelenia s nulovým priemerom a jednotkovou štandardnou odchýlkou pomocou knižnice random v jazyku Python. Na Isingovom modeli som vyberal kladný alebo záporný spin z Bernoulliho distribúcie a v Heisenbergovom modeli boli vektorý spinu vyberané pomocou unifromne náhodne generovaných uhlov ktoré udávali jedno smer.

Kovnovučná vrstva

Väčšinu modelov som trénoval na vrstve GCN (Graph Convolution network), ktorá je momentálne jedna z najpoužívanejších. Je pomerne jednoduchá na pochopenie aj na výpočet. Zaveďme notáciu pre graf $G = (V, E)$ na ktorom opíšem akutualizovanie parametrov. Nech každému $v_i \in V$ je priradený vektor h_i ktorý obsahuje skrité príznaky nernovej siet. Spojitost grafu je reprezentovaná ako matica susednosti A kde $(v_i, v_j) \in E \to A_{ij} = 1$ opačne 0. Potom ešte definujem maticu W, čo je trénovateľná matica váh, ktorú môžeme chápať ako lineárnú plne prepojenú vrstvu.

Všeobecnú aktualizáciu skrytých príznakov vrcholov s aktivačnou funkciou $ReLu$ značenou ako σ môžeme formulovať: $H' = \sigma(AHW)$. Maticové násobenie AH efektivne skombinuje informácie z okolia daného vrcholu do jedného vektoru. Liearna vrstva reprezentovaná W umožní použiť znalosti z deep learningu a vytvoriť komplexnejšie modely. Na to aby sme od všeobecnej vrstvy prešli ku GCN potrebujeme upraviť ešte upraviť 2 veci:

1. Ak v grafe nie je slučka tak sieť nebude pracovať s invormáciou v totožnom vrchole. To sa dá vyriešiť jednoducho tak, že maticu susednosti prenásobim identitou. Budem používať $\overline{A} = AI$

2. Ak budem pri trénovaní opakovane násobiť maticou susednosti, tak hodnoty môžu ľahko explodovať teda potrebujem maticu normalizovať pomocou diagonálnej $D_{ii} = \sum_{j=0} \overline{A_{ij}}$.

Finálne pravidlo na aktualizovanie skrytých váh bude

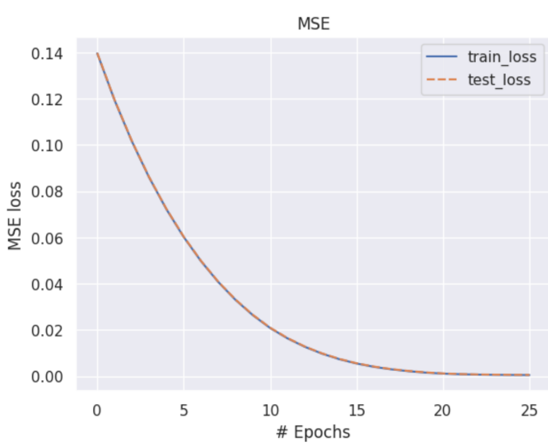
$$H' = \sigma(D^{-\frac{1}{2}}\overline{A}D^{-\frac{1}{2}}XW)$$

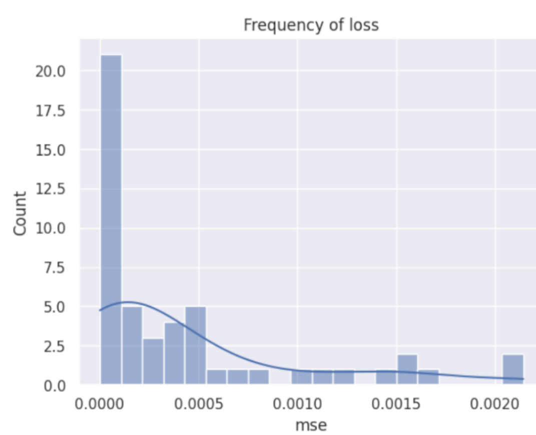
Architektúra siete

Pred tým ako spravíme návrh siete treba určiť aký typ predickie sa bude na grafe robiť. Môžeme predikovať buď nejaké informácie o vrcholoch, hranách alebo nejakú inú vlastnosť týkajúcu sa celého grafu. V tomto prípade sa jednalo o predickiu na celom grafe. Pri inferencii sa výsledok určuje tak, že na konci siete sa pridá poolingová vrstva ktorá agreguje hodnoty všetkých vrcholov. V prípade Isingového modelu bol počet vstupných príznakov 1 a pre Heisenbergov model 3, pretože spin vrcholu bol reporezentovaný 3D vektorom. V oboch pripachol mali výstupné parametre vrcholov dimenziu 1.
Všetky siete, ktoré som trénoval pozostávali s nejakého počtu konvulučných vrstiev za ktorou nasledovalo plne prepojené vrstvy a na konci poolingová vrstva. S veľkosťou siete a počatmi parametrov som už exprei-mentoval.

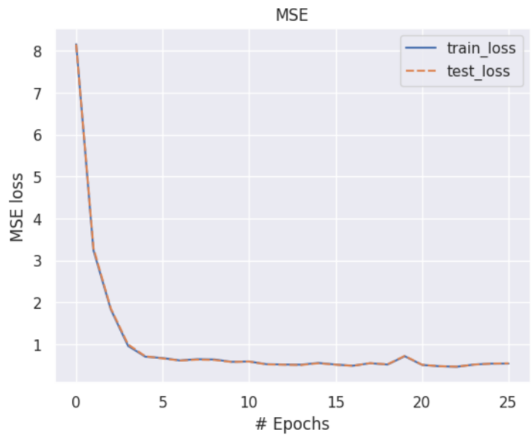
Zváver

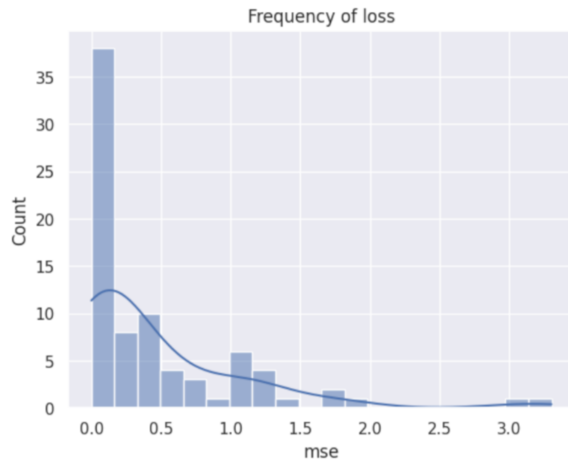
Trénovanie na Isingovom modeli s datasetom veľkosti 2500 grafov na 25 epoch trvalo na bežnom notebooku 15-20 sekúnd. Pričom loss dosahoval okolo 10^{-4}





Trénovanie na Isingovom modeli s datasetom veľkosti 1000 grafov na 25 epoch trvalo na bežnom notebooku 90 sekúnd. Pričom loss dosahoval okolo 10^{-1}


MSE
train_loss
test_loss
MSE loss
Epochs
0
5
10
15
20
25
1
2
3
4
5
6
7
8


Frequency of loss
Count
mse
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
5
10
15
20
25
30
35

Využitie