Diplomová práca

Analýza a návrh riešenia

Bc. Samuel Baran

Názov práce: Nové techniky učenia sa bez učiteľa pre klasifikáciu a predikciu molekulárnych vlastností

Vedúci práce: RNDr. Juraj Šebej, PhD.
Konzultant: RNDr. Ľubomír Antoni, PhD.

Úvod

Strojové učenie je podmnožinou umelej inteligencie, pričom sa zaoberá metódami a algoritmami učenia sa stroja z údajov. Cieľom strojového učenia je modelovanie algoritmov učenia sa pomocou stroja na základe vstupných dát v definovanom priestore riešení. Strojové učenie bolo v poslednej dobe úspešne aplikované na riešenie úloh v mnohých oblastiach vedy. V oblasti predikcie a klasifikácie molekulárnych vlastností sa stretávame s nižšou dostupnosťou údajov a vyššou heterogénnosťou vstupných údajov. Tieto skutočnosti vplývajú na nevhodnosť používania metód kontrolovaného učenia v tejto oblasti. Z týchto dôvodov vzniká priestor pre využitie metód učenia sa bez učiteľa, ktorého prednosťou je primárne využitie neoznačených dát na vytvorenie skrytých reprezentácií vstupov, ktoré poskytujú lepšiu východiskovú pozíciu pre modely učenia sa s učiteľom.

Motivácia

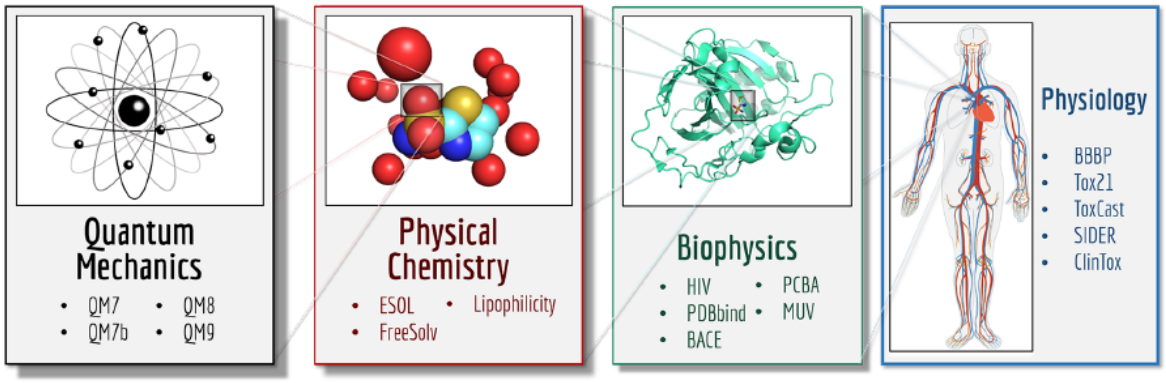
Tak ako v mnohých iných oblastiach, aj v chémii možno v posledných rokoch sledovať nárast využitia metód strojového učenia. Nové metódy v kombinácii s dostupnosťou väčších datasetov umožnili algoritmom strojového učenia uplatniť sa aj v oblasti predikcie molekulárnych vlastností.

Spočiatku bol vývoj v tejto oblasti limitovaný neprítomnosťou jednotného prístupu k vyhodnocovaniu efektívnosti jednotlivých metód. Na tento nedostatok reagovali autori článku MoleculeNet: A Benchmark for Molecular Machine Learning, ktorí v

rámci knižnice DeepChem združili viacero rôznych datasetov z oblasti predikcie molekulárnych vlastností, určili metriky vhodné na evaluáciu modelov trénovaných na týchto datasetoch a sprístupnili implementácie algoritmov vyvinutých práve pre túto oblasť.

Dáta

Dáta pre úlohy strojového učenia spracúvajúce molekuly sú vysoko heterogénne, keďže obsahujú molekuly variabilnej dĺžky pozostávajúce z rôznych navzájom prepojených komponentov. Získavanie týchto dát je vzhľadom na potrebu špecializovaných zariadení a dohľadu odborníkov náročné, čo spôsobuje, že molekulárne datasety sú oveľa menšie ako tie, ktoré sú využívané pri úlohách strojového učenia z iných oblastí.



Obr. 1: Úlohy prislúchajúce jednotlivým datasetom sa zameriavajú na rôzne kategórie molekulárnych vlastností.

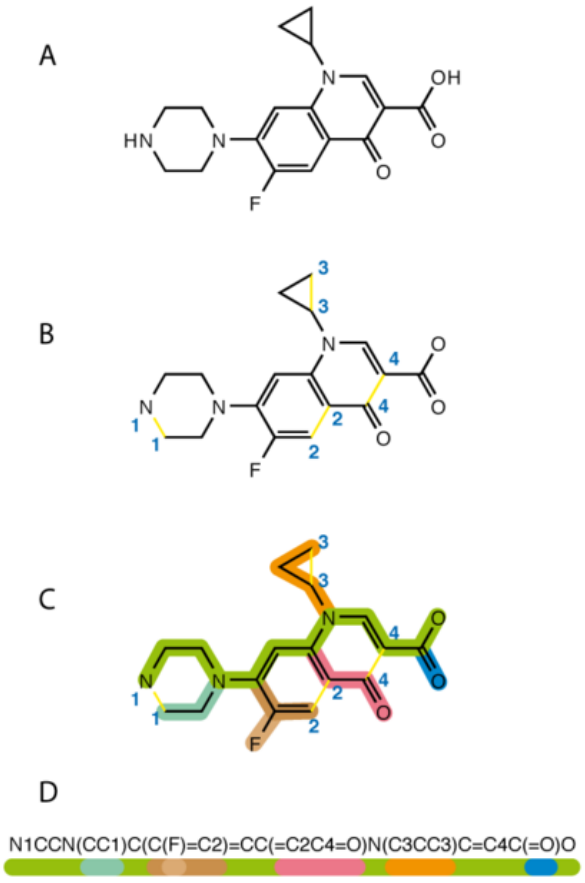
MoleculeNet obsahuje informácie o vlastnostiach vyše 700 000 chemických zlúčenín. Vzhľadom na oblasti do ktorých spadajú skúmané vlastnosti je možné rozdeliť datasety do štyroch základných kategórií: kvantová mechanika, fyzikálna chémia, biofyzika a fyziológia. Ako je znázornené na obrázku 1, jednotlivé datasety skúmajú rôzne úrovne molekulárnych vlastností, od energií excitovaných stavov (QM8), cez rozpustnosť molekúl vo vode (ESOL), až po skúmanie vlastnosti zabraňujúcej rozmnožovaniu vírusu HIV (HIV).

Reprezentácie molekúl

Väčšina datasetov využíva na kódovanie molekúl SMILES reprezentáciu, ktorá priraďuje molekulám unikátne textové reťazce variabilnej dĺžky. Vzhľadom na fakt, že väčšina metód strojového učenia požaduje vstupné dáta jednotného formátu vznikla potreba pre iné reprezentácie molekúl. V tejto kapitole si popíšeme niektoré z nich.

SMILES

SMILES notácia sa stala štandardným nástrojom na kódovanie molekúl v datasetoch. SMILES je unikátne textové označenie molekuly, ktoré vznikne linearizáciou grafu molekuly pomocou očíslovania vrcholov a hrán a následného prechádzania grafu podľa topologického usporiadania. Proces linearizácie grafu je znázornený na obrázku 2.



Obr. 2: Linearizácia grafu. V grafe sa detekujú cykly a vymažú sa hrany, ktorých odstránením sa cykly rozpoja. K názvom vrcholov hrany sa pridá jednoznačný číselný

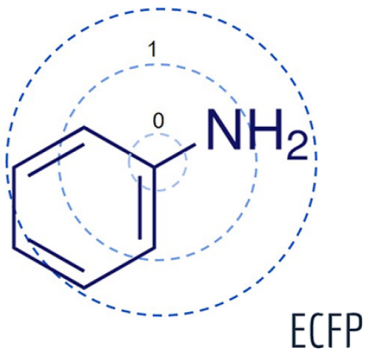
identifikátor hrany. (B) Následne sa vyberie počiatočný vrchol a generuje sa SMILES prehľadávaním do hĺbky (C).

Zo spôsobu ktorým je vytváraná SMILES reprezentácia grafu plynie, že pre jednu molekulu môže existovať viacero validných SMILES reprezentácií a ich počet rastie s komplexnosťou molekuly. Platí však, že každý validný SMILES kóduje práve jednu molekulu.

Za účelom dosiahnutia bijektívneho zobrazenia z množiny molekúl do množiny ich SMILES reprezentácií boli vyvinuté algoritmy, ktoré pre každú molekulu vyberajú jednu reprezentáciu nazývanú kanonický SMILES. Neexistuje však globálny kanonický SMILES, keďže jednotlivé nástroje využívajú rôzne kanonizačné algoritmy.

ECFP

ECFP (Extended-Connectivity Fingerprints) je široko používaný typ kódovania molekúl v chemoinformatike. Proces kódovania je rozdelený do viacerých fáz. V prvej fáze sa z molekuly odstránia atómy vodíka. Následne sa každému atómu priradí kladný celočíselný identifikátor, ktorý vznikne zahešovaním základných vlastností atómu, ako napríklad atómové číslo alebo atómová hmotnosť. V ďalšej iteratívnej fáze sa na kódovaní atómov podieľajú heše atómov z jeho okolia, ktorého veľkosť je definovaná poradím iterácie (Obr. 3).



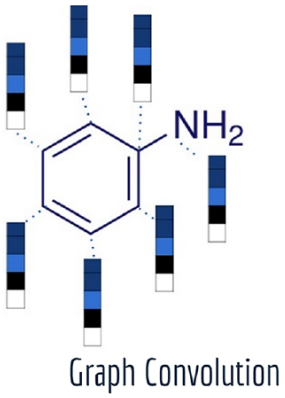
Obr. 3: Náznak procesu ECFP kódovania molekuly.

Poslednou fázou je zlúčenie všetkých kódov atómov do vektora bitov fixnej dĺžky a to tak, že sa vypočítajú zvyšky po delení kódov atómov dĺžkou vektora a vo finálnom

vektore sa na pozície zodpovedajúce zvyškom priradia jednotky. Ostatné pozície budú nastavené na hodnotu 0. Pri ECFP kódovaní môže nastať kolízia bitov, teda že dva rôzne kódy atómov majú rovnaký zvyšok po delení, čím nastáva strata informácie. Tento aspekt sa dá ovplyvniť veľkosťou finálnej reprezentácie.

Graph convolutions

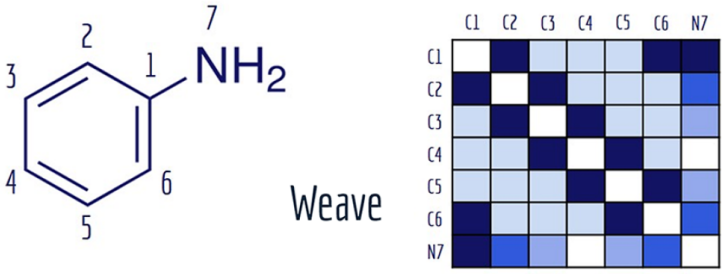
Reprezentácia pomocou grafovej konvolúcie kóduje informácie o vlastnostiach jednotlivých atómov molekuly a o ich vzájomnom prepojení. (Obr. 4) Každému atómu v molekule prislúcha vektor, v ktorom sú združené základné vlastnosti popisujúce jeho lokálne chemické prostredie. Medzi tieto vlastnosti patria napríklad typ atómu, typ hybridizácie alebo štruktúra valenčných vrstiev. Prepojenie atómov je reprezentované pomocou zoznamu hrán alebo susedov. Takáto reprezentácia je vhodná pre väčšinu grafových modelov.



Obr. 4: Vizualizácia reprezentácie molekuly pomocou grafovej konvolúcie.

Weave

Podobne ako grafová konvolúcia aj reprezentácia nazývaná Weave kóduje lokálne chemické prostredie aj vzájomné prepojenie atómov. Vektory popisujúce vlastnosti atómov sú presne také isté, no kódovanie konektivity atómov v rámci molekuly je detailnejšie ako v predošlom prípade. Namiesto zoznamu susedov Weave formát počíta vektor pre každú dvojicu atómov v molekule, ktorý obsahuje informácie o priamom vzájomnom prepojení daných atómov, type prepojenia alebo vzájomnej polohe, čím vytvára maticu vektorov. (Obr. 5) Weave reprezentácia je vhodná pre grafové modely, ktoré podporujú výpočty zahŕňajúce vlastnosti atómov a ich väzieb.



Obr. 5: Vizualizácia Weave reprezentácie molekuly.

Učenie s učiteľom v oblasti spracovania molekúl

S nástupom sofistikovaných metód hlbokého učenia zaznamenalo strojové učenie zvýšenú pozornosť širokej vedeckej komunity. Data driven analýzy sa stali rutinným krokom v mnohých chemických a biologických aplikáciách, akými sú napríklad virtuálny skríning, predikcie chemických vlastností alebo výpočty v oblasti kvantovej chémie. V tejto kapitole si popíšeme algoritmy strojového učenia, ktoré boli úspešne aplikované na úlohy predikcie molekulárnych vlastností.

Metóda podporných vektorov

Metóda podporných vektorov (SVM) patrí k jednej z najpoužívanejších metód strojového učenia. SVM algoritmus hľadá nadrovinu priestoru príkladov, ktorá oddeľuje triedy príkladov v danom priestore. Takýchto nadrovín môže byť viacero, a preto cieľom algoritmu je nájsť takú nadrovinu, ktorá rozdeľuje príklady do kategórií tak, že vzdialenosť najbližších reprezentantov jednotlivých tried je maximálna možná. V prípadoch kedy nie je možné lineárne separovať triedy, SVM algoritmus ponúka možnosť použiť kernel funkcie na zlepšenie úspešnosti modelu.

Náhodné lesy

Náhodné lesy (RF) patria ku skupinovým modelom strojového učenia. Pozostávajú z niekoľkých samostatných rozhodovacích stromov (DT), ktoré sú trénované na náhodných podmnožinách dátových atribútov. Výstupy týchto stromov sú následne agregované a prezentované ako výstup metódy RF. RF sú použiteľné na regresné a klasifikačné úlohy.

Gradient Boosting

Ďalším skupinovým modelom, ktorý tvoria individuálne rozhodovacie stromy, je Gradient Boosting. Narozdiel od RF, táto metóda pozostáva z jednoduchých DT a je budovaná sekvenčne. V každom kroku je nový strom generovaný greedy spôsobom a je obmedzený buď počtom uzlov, listov alebo hĺbkou stromu.

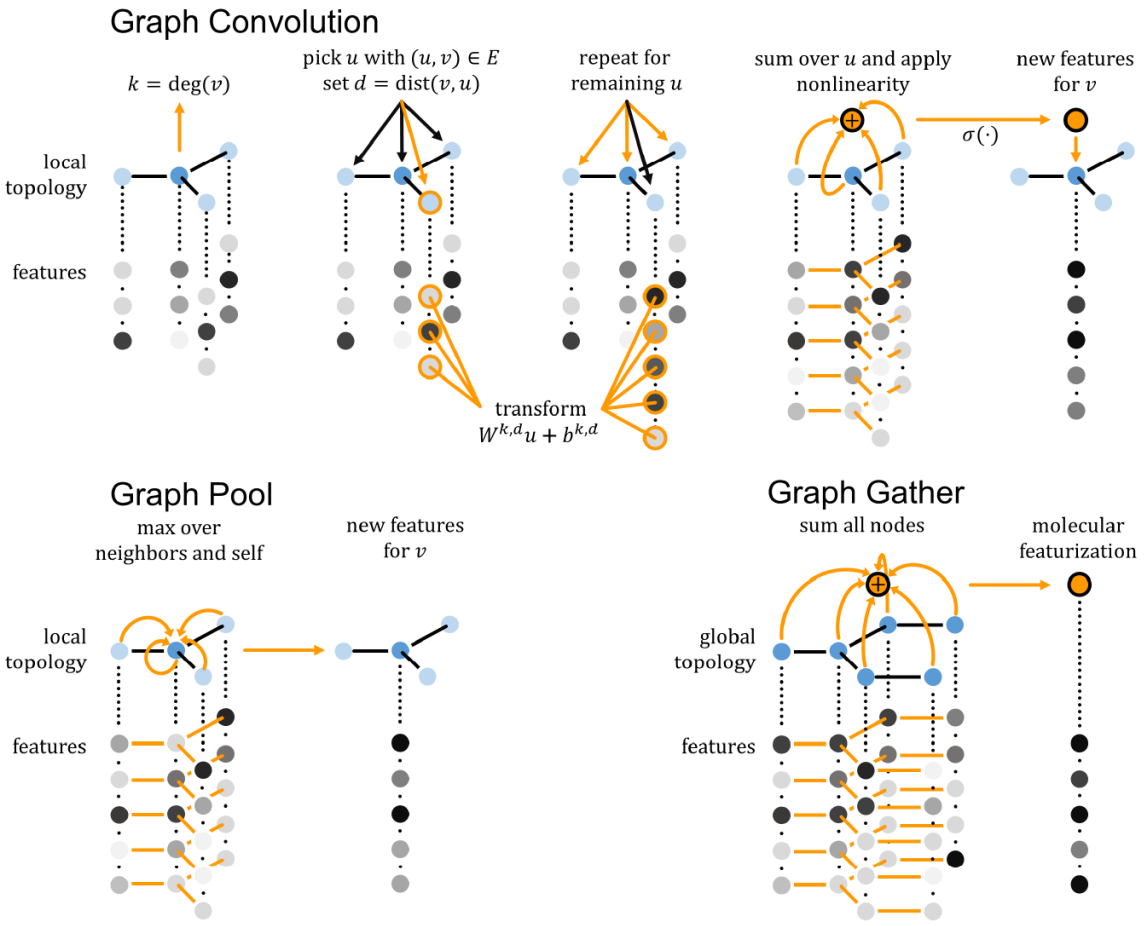
GCN

Vzhľadom na štruktúru všeobecnej molekuly a vzájomnú interakciu atómov, jej stavebných zložiek, sú grafy vhodnou reprezentáciou, ktorá dokáže zachytiť veľké množstvo informácií. V poslednej dekáde boli vyvinuté viaceré grafové neurónové siete (GNN), ktorých špecifikom bolo priame spracovanie grafu ako vstupnej informácie.

Jedným z príkladov je grafová konvolučná sieť (GCN) predstavená v článku Convolutional Networks on Graph for Learning Molecular Fingerprints. Táto sieť pozostáva z grafových konvolučných vrstiev, ktoré sú zovšeobecnením dvojdimenzionálnej konvolúcie používanej pri spracovaní obrazu. V štandardných konvolučných sieťach sa nahliada na obrázok ako na mriežku pixelov. Na pixely z blízkeho okolia centrálneho pixelu je aplikovaná plne prepojená vrstva nazývaná konvolučný filter, ktorá počíta výstup konvolúcie. Podobne je tomu aj pri počítaní konvolúcie pre konkrétny vrchol grafu, kde je na vektory reprezentujúce jeho susedné vrcholy aplikovaná plne prepojená vrstva vytvárajúca novú reprezentáciu daného vrcholu. (Obr. 6) Princíp konvolúcie v oboch prípadoch umne využíva lokálnu geometriu systému na to, aby sa zredukoval počet učiacich sa parametrov siete.

V klasických konvolučných sieťach sú tiež konvolučné vrstvy nasledované poolingovými vrstvami, ktoré redukujú dimenziu dát pomocou agregačných funkcií min, max alebo avg. Aj v GCN je analogicky definovaná poolingová vrstva, ktorá aplikuje jeden z agregačných operátorov na vrchol a jeho susedov. (Obr. 6)

Každému vrcholu v tomto grafovo-konvolučnom systéme prislúcha vektor popisujúcich hodnôt, avšak pri úlohách predikcie vlastnosti grafu je potrebné mať jeden vektor fixnej dĺžky reprezentujúci celý graf. Na tento účel bola navrhnutá operácia gather združujúca vektory všetkých vrcholov pomocou sčítania. (Obr. 6)



Obr. 6: Grafická reprezentácia základných grafových operácií použitých v GCN. Operácie sú znázornené na jednom vrchole. Každá operácia, ktorá je aplikovaná na vrcholy tmavomodrej farby, nemení vrcholy svetlomodrej farby. Grafová konvolúcia a grafová pooling vrstva sú znázornené len na jednom vrchole a jeho susedstve, ale tieto operácie sú vykonávané súčasne na všetkých vrcholoch.

Samokontrolované učenie v oblasti spracovania molekúl

Ako sme vyššie spomínali, pri úlohách predikcie molekulárnych vlastností sa stretávame so zhoršenou dostupnosťou anotovaných dát, čo znemožňuje efektívne využitie metód učenia sa s učiteľom, keďže modely trénované na takto malých datasetoch zle generalizujú a sú náchylné na preučenie. Preto sa v poslednej dobe vývoj zameral aj na oblasť samokontrolovaného učenia, ktorého prednosťou je primárne využitie neoznačených dát na vytvorenie skrytých reprezentácií vstupov,

ktoré poskytujú lepšiu východiskovú pozíciu pre modely učenia sa s učiteľom. Predstavíme si dva takéto prístupy: ChemBERTa a MolCLR.

ChemBERTa

Väčšina konvenčných metód strojového učenia vyžaduje vstup rovnakej dĺžky, čo je pri SMILES reprezentácii molekúl vzhľadom na ich variabilnú veľkosť netriviálna úloha.

Hlavnými prístupmi pri riešení tohto problému je využitie grafových neurónových sietí (GNN) a metóda chemických odtlačkov (chemical fingerprints). Vzhľadom na fakt, že sa transformery stali štandardným nástrojom na učenie reprezentácií v oblasti spracovania prirodzeného jazyka (NLP), vyvstala otázka na preskúmanie prínosu tejto metódy aj pri spracovávaní textových reprezentácií chemických zlúčenín. Práca ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction ukázala, že transformery sú minimálne konkurencieschopným nástrojom využiteľným pri predikcii molekulárnych vlastností.

Algoritmus ChemBerta je založený na implementácii modelu RoBERTa, ktorý vylepšil samokontrolovaný predtréning transformerov nazvaný BERT. Tieto modely si pre lepšie pochopenie modelu ChemBERTa v krátkosti popíšeme.

BERT je obojsmerný transformer učiaci sa jazykovú reprezentáciu tréningom na veľkom množstve neoznačených textových dát. Princíp učenia je postavený na kombinácii dvoch generatívnych metód: maskovaní, následnej predikcii náhodne maskovaných tokenov (MLM - Masked Language Model) a na predikcii nasledujúcej vety (NSP - Next Sentence Prediction).

Model RoBERTa (Robustly optimized BERT approach) vznikol opätovným natrénovaním modelu BERT s použitím 10 násobne väčšieho objemu dát a vylepšením metodológie tréningu tým, že sa zaviedlo dynamické maskovanie tokenov, výrazne sa zvýšila veľkosť dávok a odstránil sa komponent predikujúci poradie viet.

V rámci evaluácie bola metóda ChemBERTa predtrénovaná na datasete obsahujúcom 10 miliónov chemických zlúčenín porovnávaná s najpoužívanejšími modelmi v oblasti predikcie molekulárnych vlastností, ktorými sú náhodné lesy (RF), metóda podporných vektorov (SVM) a orientovaná neurónová sieť s preposielaním správ (directed message passing neural network - D-MPNN). Výsledky porovnaní sú zhrnuté v tabuľke 1.

Tab. 1: Porovnanie modelu ChemBERTa predtrénovaného na 10 miliónovom datasete PubChem a úspešných modelov na vybraných úlohách predikcie molekulárnych vlastností

	BBBP 2,039		**ClinTox (CT_TOX)** 1,478		**HIV** 41,127		**Tox21 (SR-p53)** 7,831	
	ROC	PRC	ROC	PRC	ROC	PRC	ROC	PRC
ChemBERTa 10M	0.643	0.620	0.733	0.975	0.622	0.119	**0.728**	0.207
D-MPNN	**0.708**	0.697	**0.906**	**0.993**	0.752	0.152	0.688	**0.429**
RF	0.681	0.692	0.693	0.968	**0.780**	**0.383**	0.724	0.335
SVM	0.702	**0.724**	0.833	0.986	0.763	0.364	0.708	0.345

Aj keď ChemBERTa neprekonal vybrané modely, ukázali transformery potenciál pre ďalšie využitie v tejto oblasti, jednak tým že sú dobre škálovateľné vzhľadom na veľkosť datasetu určeného na predtréning, ale aj možnosťou využitia vizualizácie attention mechanizmu.

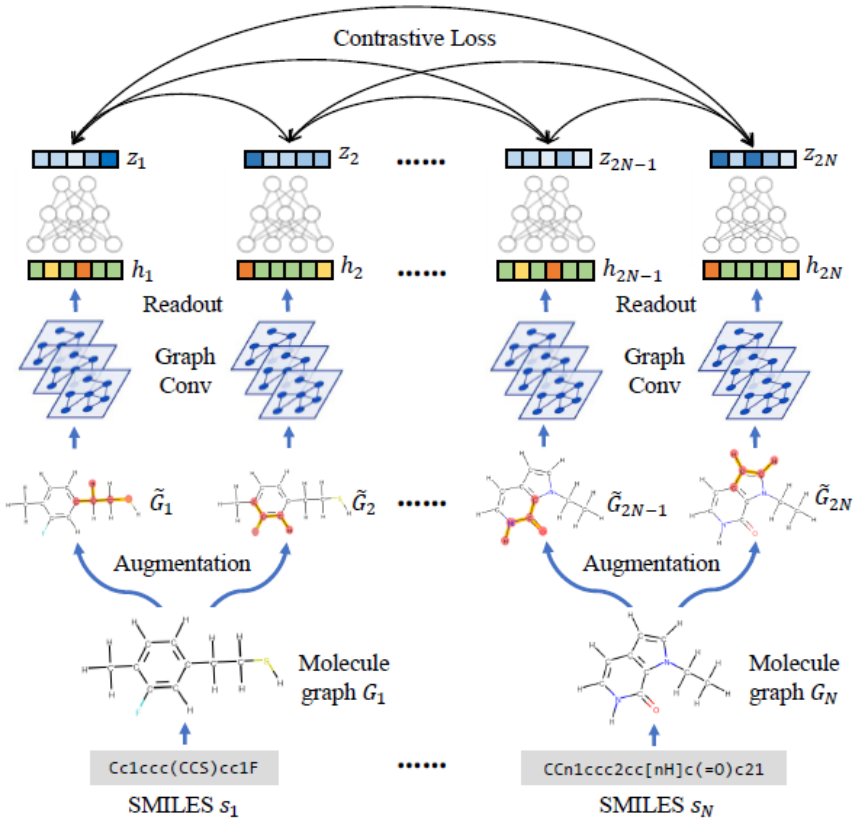
MolCLR

Jedným z ďalších prístupov samokontrolovaného učenia je kontrastívne učenie, kde sú reprezentácie vstupu trénované pomocou augmentácií vstupu a následnej snahy dosiahnuť rovnaké reprezentácie pre takto augmentované vstupy.

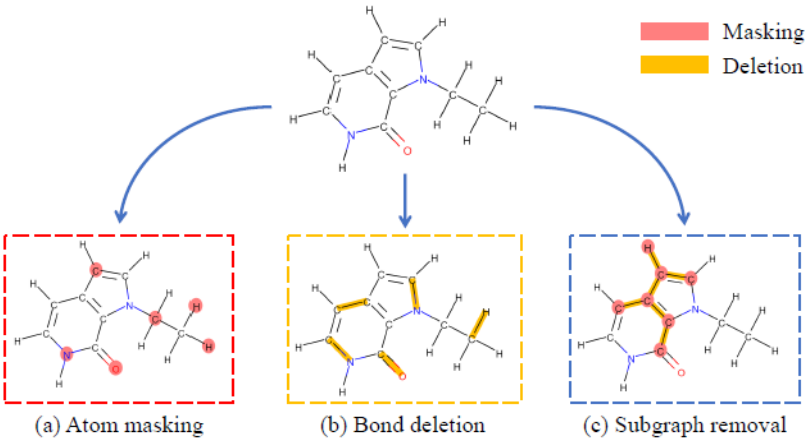
Využitím grafových neurónových sietí a aplikovaním kontrastívneho predtréningu vznikol model MolCLR: Molecular Contrastive Learning of Representations via Graph Neural Networks pomenovaný podľa jedného z prvých modelov kontrastívneho učenia SimCLR. Architektúru modelu opisuje obrázok 7.

GNN sú v tomto prípade využívané na kódovanie grafu molekúl a oproti SMILES reprezentácii sú schopné uchovávať aj informácie o topológii molekúl a ďalších vlastnostiach jednotlivých atómov a chemických väzieb.

Aby bolo možné využiť princípy kontrastívneho učenia, boli predstavené 3 stratégie augmentácie aplikovateľné na GNN: maskovanie atómu, odstránenie väzieb a odstránenie podgrafu. Pri maskovaní atómu je vopred určený počet náhodných atómov nahradený špeciálnym znakom. Odstraňovanie väzieb narozdiel od maskovania atómov odstraňuje náhodne vybrané chemické väzby. Na odstránenie podgrafu je možné nahliadať ako na kombináciu predchádzajúcich dvoch augmentácií. Vizualizáciu týchto augmentácií zachytáva obrázok 8.



Obr. 7: Molekulárne kontrastívne učenie reprezentácií s použitím GNN. SMILES reprezentácia s_i je transformovaná na graf molekuly G_i ,na ktorý sú následne aplikované dva náhodne vybrané grafové augmentácie. Tým dostávame dva augmentované grafy G'_{2i-1}, G'_{2i}, ktoré sú vstupom pre sieť počítajúcu reprezentácie molekúl. Na takto vypočítané reprezentácie je aplikovaná kontrastívna stratová funkcia s cieľom maximalizovať zhodu reprezentácií dvojíc augmentácií.



Obr. 8: Tri augmentácie grafu prislúchajúceho molekule. (a) Maskovanie atómov náhodne nahradí vybrané atómy maskovacou značkou. (b) Odstránenie väzby náhodne odstráni väzby medzi dvojicami atómov. (c) Odstránenie podgrafu náhodne odstráni indukovaný podgraf.

Experimenty zachytené v tabuľke 2 ukázali, že metóde MolCLR sa na viacerých úlohách predikcie vlastností molekúl podarilo prekonať najlepšie modely z oblasti kontrolovaného učenia, akými sú náhodné lesy (RF), metóda podporných vektorov (SVM), orientovaná neurónová sieť s preposielaním správ (directed message passing neural network - D-MPNN) a grafová neurónová sieť MGCNN vyvinutá práve na predikciu molekulárnych vlastností.

Tab. 2: Porovnanie modelov na základe ROC-AUC (%) testovacej vzorky, kde prvé štyri modely sú modely učenia s učiteľom a zvyšné tri sú modely samokontrolovaného učenia.

Dataset	BBBP	Tox21	ClinTox	HIV	BACE	SIDER	MUV
# Molecules	2039	7831	1478	41127	1513	1478	93087
# Tasks	1	12	2	1	1	27	17
RF	71.4±0.0	76.9±1.5	71.3±5.6	78.1±0.6	**86.7±0.8**	**68.4±0.9**	63.2±2.3
SVM	72.9±0.0	**81.8±1.0**	66.9±9.2	**79.2±0.0**	86.2±0.0	**68.2±1.3**	67.3±1.3
MGCN [74]	**85.0±6.4**	70.7±1.6	63.4±4.2	73.8±1.6	73.4±3.0	55.2±1.8	70.2±3.4
D-MPNN [28]	71.2±3.8	68.9±1.3	**90.5±5.3**	75.0±2.1	85.3±5.3	63.2±2.3	**76.2±2.8**
HU. et.al [60]	70.8±1.5	78.7±0.4	78.9±2.4	80.2±0.9	85.9±0.8	65.2±0.9	81.4±2.0
N-Gram [75]	**91.2±3.0**	76.9±2.7	85.5±3.7	**83.0±1.3**	87.6±3.5	63.2±0.5	81.6±1.9
MolCLR	73.6±0.5	**79.8±0.7**	**93.2±1.7**	80.6±1.1	**89.0±0.3**	**68.0±1.1**	**88.6±2.2**

Model zaznamenal lepšie výsledky aj oproti metódam samokontrolovaného učenia, kde bol porovnávaný s modelmi grafových neurónových sietí navrhnutými Hu a kolektívom a tiež aj s modelom generujúcim reprezentácie molekúl pomocou

N-gramového grafu (N-gram graph: Simple unsupervised representation for graphs, with applications to molecules).

Samokontrolované učenie v oblasti počítačového videnia

Oblasť počítačového videnia sa venuje viacerým úlohám, ako sú napríklad klasifikácia obrázkov, detekcia objektov alebo segmentácia obrazu.

Pri úlohách spracovania obrazu sa osvedčili konvolučné neurónové siete, avšak aj tie vzhľadom na dimenzionalitu a rozmer vstupov potrebujú veľké datasety, aby predišli preučeniu. Ako alternatíva sa ukázalo využitie transformerov inšpirované úspešnými aplikáciami v oblasti spracovania prirodzeného jazyka (NLP), čo vyústilo do modelu nazývaného visual transformer (ViT). Aj napriek tomu, že modely ViT konkurujú konvolučným sieťam, nepodarilo sa preukázať jednoznačné prednosti týchto modelov. ViT vyžadujú viac tréningových dát a sú výpočtovo náročnejšie.

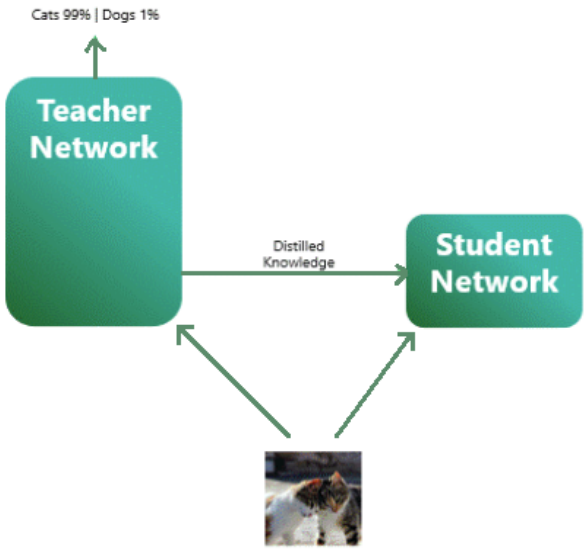
Vzhľadom na náročnosť manuálnej anotácie veľkých množstiev vstupných dát aj v tejto oblasti vznikla potreba využitia samokontrolovaných metód strojového učenia. Tieto metódy ukázali potenciál v spojení s konvolučnými neurónovými sieťami, kde príkladom môže byť SimCLR, jedna z metód kontrastívneho učenia alebo samokontrolované metódy učenia sa reprezentácií obrázkov MoCo (Momentum Contrast for Unsupervised Visual Representation Learning) a BOYL (Bootstrap Your Own Latent).

Úspech nekontrolovaného predtréningu transformerov v oblasti NLP (BERT, GPT) bol motiváciou pre prispôsobenie a aplikáciu týchto princípov aj v oblasti počítačového videnia. Článok Generative pretraining from pixels opisuje model učiaci sa reprezentácie obrazu vhodné na ďalšie úlohy predikcie a klasifikácie, pomocou generatívneho predikovania časti obrazu. Ďalším prístupom je metóda DINO, ktorej sa budeme venovať v nasledujúcej kapitole.

Dino

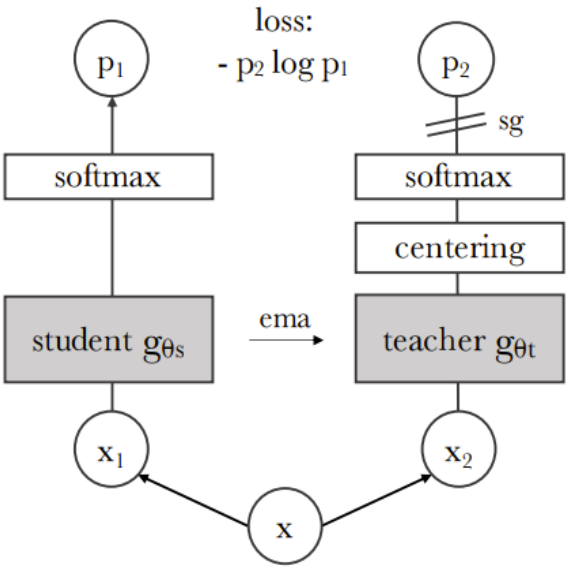
DINO (self distilation with no labels) je samokontrolovaná metóda učenia sa reprezentácií obrazu predstavená v práci Emerging Properties in Self-Supervised Vision Transformers, ktorá ako základ používa princíp odvodený z destilácie znalostí.

Destilácia znalostí (Obr. 9) je proces, do ktorého vstupuje veľký, nie nutne označkovaný dataset a dve siete: učiteľská, zvyčajne zložitejšia so znalosťami nadobudnutými predchádzajúcim tréningom na nejakej konkrétnej úlohe a netrénovaná študentská s jednoduchšou štruktúrou. Myšlienka destilácie znalostí spočíva v tom, že predikcie získané pomocou učiteľskej siete sú prezentované študentskej sieti ako značky, ktoré má predikovať a na základe ktorých sa učí. Výsledkom tohto procesu je študentská sieť s podobnými znalosťami ako učiteľská.



Obr. 9: Destilácia znalostí. Vstupné dáta sú poskytnuté na vstup obom sieťam a študentská sieť sa učí predikovať to čo predikuje učiteľská.

Metóda DINO pozostáva z dvoch sietí s názvami študentská a učiteľská, ktoré majú rovnakú architektúru, ale nezdieľajú učiace sa parametre inak nazývané váhy. Na tieto siete (Obr. 10) sa nahliada ako na destiláciu, počas ktorej sa študentská sieť učí na základe predikcií učiteľskej siete.



Obr. 10: Architektúra DINO. Algoritmus dá na vstup študentskej a učiteľskej siete náhodné transformácie vstupného obrázku. Obe siete majú rovnakú architektúru, ale rozličné parametre. Výstup učiteľskej siete je centralizovaný pomocou priemeru počítaného naprieč celou dávkou (batch). Výstupy sietí sú normalizované pomocou softmax funkcie s parametrom teploty a ich podobnosť je vypočítaná pomocou krížovej entropie. Na konci učiteľskej siete je operátor zabraňujúci spätnému šíreniu gradientu, čo spôsobuje, že chyba je spätne šírená len študentskou sieťou. Parametre učiteľskej siete sú zmenené pomocou ema funkcie (exponential moving average).

Rozdiel oproti klasickej destilácii je v tom, že ani učiteľská sieť nie je natrénovaná na konkrétnej úlohe, ale počas tréningu sa časť vedomostí nadobudnutých študentskou sieťou v predchádzajúcich iteráciach prenáša na učiteľskú.

Ďalším podstatným rozdielom je spôsob práce s dátami. Oproti klasickej destilácii v ktorej sú obom sieťam poskytované rovnaké tréningové príklady, sú v metóde DINO vytvárané lokálne a globálne výseky toho istého obrázka, ktoré sú následne prerozdelené sieťam. Globálne výseky sú definované ako výseky zaberajúce viac ako polovicu pôvodného obrázka, zatiaľ čo lokálne majú menší rozmer. (Obr. 11) Študentská sieť počíta výstup pre dva globálne a niekoľko ďalších lokálnych výsekov, zatiaľ čo učiteľská sieť pracuje len s dvoma globálnymi výsekmi.

Obr. 11: Príklad globálnych a lokálnych výsekov obrázkov.

Hodnoty predikované učiteľskou sieťou sú centralizované pomocou kumulovanej hodnoty priemeru výstupov siete počítaných naprieč dávkou. Následne sú výstupy oboch sietí normalizované aplikovaním funkcie softmax s parametrom teploty, ktorý zabezpečuje vyostrenie predikcií. Oba tieto princípy sú zachytené v algoritme 1.

Algoritmus 1: Pseudokód metódy DINO

```
# gs, gt: student and teacher networks
# C: center (K)
# tps, tpt: student and teacher temperatures
# l, m: network and center momentum rates
gt.params = gs.params
for x in loader: # load a minibatch x with n samples
    x1, x2 = augment(x), augment(x) # random views

    s1, s2 = gs(x1), gs(x2) # student output n-by-K
    t1, t2 = gt(x1), gt(x2) # teacher output n-by-K

    loss = H(t1, s2)/2 + H(t2, s1)/2
    loss.backward() # back-propagate

    # student, teacher and center updates
    update(gs) # SGD
    gt.params = l*gt.params + (1-l)*gs.params
    C = m*C + (1-m)*cat([t1, t2]).mean(dim=0)

def H(t, s):
    t = t.detach() # stop gradient
    s = softmax(s / tps, dim=1)
    t = softmax((t - C) / tpt, dim=1) # center + sharpen
    return - (t * log(s)).sum(dim=1).mean()
```

Autori algoritmu DINO tiež predstavili vlastnú chybovú funkciu, ktorá kombinuje výstupy sietí počítané z viacerých globálnych a lokálnych výsekov toho istého obrázka. Čiastočná chyba medzi dvoma výstupmi študentskej a učiteľskej siete je počítaná pomocou krížovej entropie, kde je ako očakávaný výstup použitá predikcia učiteľskej siete. Výpočet čiastkovej chyby je aplikovaný na všetky dvojice výstupov študentskej a učiteľskej siete okrem tých, ktoré boli počítané z rovnakého výseku. Celková chyba je potom definovaná ako priemer týchto hodnôt.
Nech L je množina lokálnych výsekov a G je množina globálnych výsekov pre nejaký konkrétny obrázok. Označme množinu dvojíc vstupov pre študentskú a učiteľskú sieť ako:

$$C = (L \cup G) \times G) \setminus \{(c, c) | \, c \in G\}.$$

Potom celkovú chybu siete pre výpočet na danom obrázku vyjadruje vzťah:

$$Dino\ LOSS = \frac{1}{|C|} \sum_{(c_s, c_t) \in C} H(P_t(c_t), P_s(c_s)),$$

kde $P_t(c_t)$ je výstup učiteľskej siete pre vstup c_t, $P_s(c_s)$ je výstup študentskej siete pre vstup c_s a H je krížová entropia definovaná pre vstupy dĺžky n nasledovne:

$$H(a, b) = \sum_{i=0}^{n-1} - a_i \log b_i .$$

Narozdiel od študentskej siete, ktorej váhy sú modifikované pomocou algoritmu spätného šírenia chyby, sa učiteľská sieť učí pomocou EMA metódy (exponential moving average):

$$W_{teacher} = m * W_{teacher} + (1 - m) * W_{student},$$

kde $W_{teacher}$ a $W_{student}$ sú váhy učiteľskej a študentskej siete a m je moment učenia.

Samotná architektúra sietí pozostáva z dvoch komponentov: kostry, ktorú tvorí buď konvolučná sieť alebo visual transformer a projekčnej hlavy. Po natrénovaní siete sa ako reprezentácie obrázkov berú výstupy kostry siete.

Takto navrhnutá architektúra produkuje reprezentácie obrázkov, ktoré v sebe nesú explicitnú informáciu o sémantickej segmentácii obrazu a tiež sú vhodné ako vstup pre KNN algoritmus.

Návrh riešenia

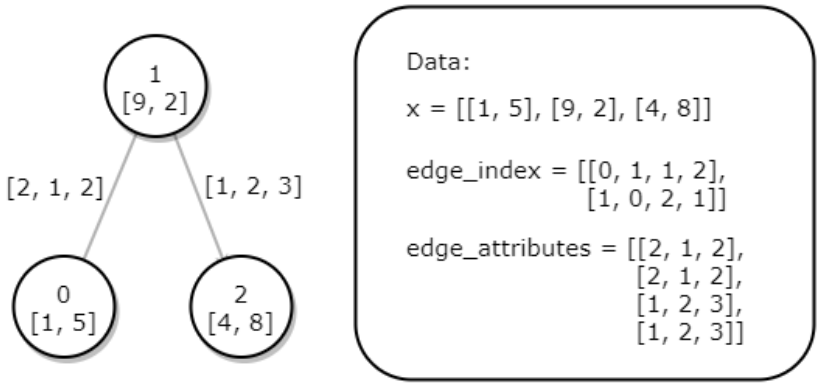
Jedným z cieľov práce je návrh metódy klasifikácie a predikcie molekulárnych vlastností inšpirovanej najnovšími technikami učenia sa bez učiteľa z oblasti počítačového videnia. Rozhodli sme sa využiť myšlienku samokontrolovaného tréningu DINO a aplikovať ju v chemickej doméne na spracovanie grafových reprezentácií molekúl.
V ďalších podkapitolách si presnejšie popíšeme reprezentáciu molekúl, transformácie týchto reprezentácií, ich dávkovanie, návrh neurónovej siete a návrh učiaceho algoritmu.

Reprezentácia molekúl
Prvá z potrebných zmien súvisela s prácou s dátami. Vzhľadom na fakt, že sme pracovali s molekulami namiesto obrazových dát, bolo potrebné vybrať vhodné reprezentácie molekúl, ktoré by umožňovali vytváranie globálnych a lokálnych výsekov tak, aby bola zachovaná funkčná podstata tvorby týchto výsekov. Mnohé reprezentácie dát boli priamo asociované s algoritmami, pre ktoré boli vytvárané, preto bolo do rozhodovania potrebné zahrnúť aj tento aspekt.
Do úvahy prichádzala možnosť priameho použitia textovej reprezentácie SMILES a nahradenia ViT v DINO architektúre klasickými transformermi.

Myšlienka nahliadať na SMILES ako na text jazyka pozostávajúci zo základných znakov abecedy spájaných na základe gramatických pravidiel, čím sa prinavracia transformerom ich pôvodný význam v oblasti NLP, bola použitá v modeli ChemBERTa. Skúmali sme možnosť tvorby náhodných výsekov na týchto reprezentáciách, ale vzhľadom na spôsob akým sú dané reprezentácie vytvárané, nebolo možné zaručiť celistvosť vytvorených výsekov. Čisto textová reprezentácia molekuly je tiež ochudobnená o doménové znalosti o atómoch tvoriacich molekulu a aj o väzbách definujúcich štruktúru molekúl.

Ďalším prístupom bola reprezentácia molekúl vo forme grafov a následné spracovanie grafovými neurónovými sieťami GNN. Grafová reprezentácia, narozdiel od textovej, ponúka možnosť doplniť vlastnosti vrcholov a hrán grafu. Táto reprezentácia bola použitá pri tréningu viacerých úspešných modelov ako napríklad GCN (Graph Convolution Network), MPNN (Message Passing Neural Network), D-MPNN (Directed Message Passing Neural Network) ale aj pri metóde MolCLR.

Pri tvorbe nášho modelu sme sa rozhodli použiť grafovú reprezentáciu molekúl, do ktorej sme zahrnuli atómové čísla, priestorové rozmiestnenie väzieb atómu ako atribúty atómov, typy a orientácie jednotlivých väzieb ako atribúty pre väzby. Prvotný graf molekuly spolu s vlastnosťami sme zo SMILES reprezentácie získali pomocou knižnice RDKit. Ten bol následne prevedený na reprezentáciu vhodnú pre spracovanie grafovou neurónovou sieťou implementovanou pomocou triedy Data z knižnice PyG.

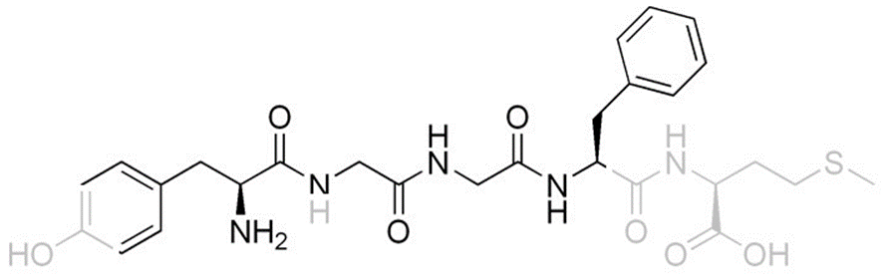


Obr. 12: Neorientovaný graf a jeho reprezentácia pomocou triedy Data

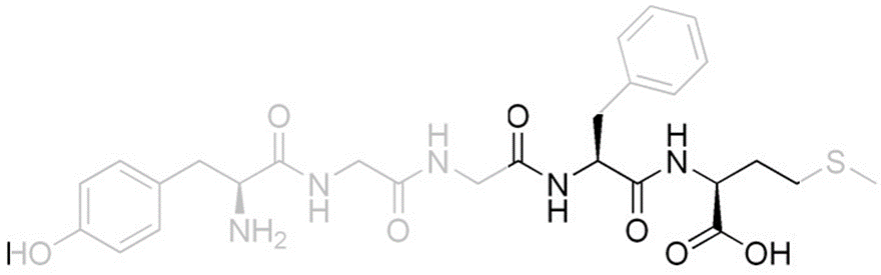
Trieda Data udržiava graf pomocou viacerých premenných: `x`, `edge_index` a `edge_attributes`, kde premenná `x` je tenzor tvaru `(počet vrcholov, počet vrcholových atribútov)`, ktorá obsahuje atribúty pre jednotlivé vrcholy, tenzor s veľkosťou `(2, počet hrán)` prislúchajúci premennej `edge_index` kóduje hrany grafu a premenná `edge_attributes` rozmerov `(počet hrán, počet atribútov hrán)` sprístupňuje atribúty jednotlivých hrán.

Transformácie reprezentácií molekúl

Metóda DINO stojí na myšlienke generovania náhodných výsekov vstupných obrázkov. Aby bol tento princíp aplikovateľný aj na naše dáta, potrebovali sme vymyslieť spôsob, akým robiť výseky molekúl. Vzhľadom na to, že naše vstupné dáta sú reprezentované grafom, rozhodli sme sa generovať náhodné súvislé podgrafy pôvodného grafu. Obrázky 13 a 14 postupne zobrazujú globálny a lokálny výsek molekuly s naznačením grafu pôvodnej molekuly.



Obr. 13: Globálny výsek (čiernou) vyznačený v pôvodnom grafe molekuly (sivou)



Obr.14: Lokálny výsek (čiernou) vyznačený v pôvodnom grafe molekuly (sivou)

Na generovanie takýchto podgrafov sme skonštruovali randomizovaný algoritmus (Algoritmus 2). Vstupom algoritmu je graf molekuly a začiatok a koniec intervalu limitujúceho pomer počtu vrcholov podgrafu k počtu vrcholov pôvodného grafu. Tento interval je vopred definovaný samostatne pre lokálne aj globálne výseky. Počet vrcholov podgrafu je preto určený náhodne zvoleným pomerom z tohto intervalu. Množinu vrcholov, ktorá bude slúžiť na vytvorenie indukovaného podgrafu pôvodného grafu molekuly, sme inicializovali náhodne vybraným centrálnym atómom a následne sme ju iteratívne rozširovali tak, aby graf indukovaný touto množinou vrcholov bol v každom kroku spojitý, až kým množina nenadobudne vopred zvolenú veľkosť. Pri implementácii sme využili Python knižnicu NetworkX určenú na

vytváranie, modifikáciu a štúdium štruktúry, vlastností a funkcií komplexných grafov a sietí.

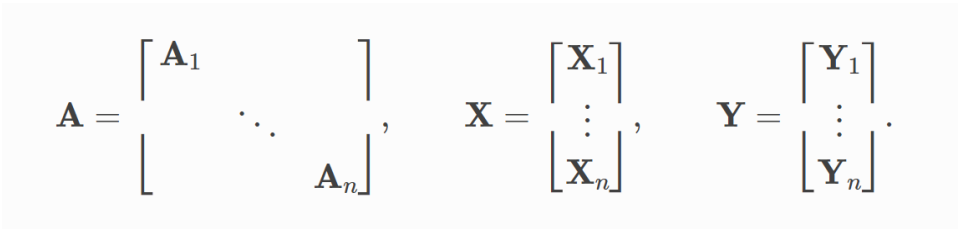
Algoritmus 2: Pseudokód metódy na generovanie náhodného spojitého podgrafu grafu reprezentujúceho molekulu na základe intervalu, ktorý vymedzuje pomer počtu vrcholov v podgrafe a v pôvodnom grafe.

```
get_random_subgraph(G, min_fraction, max_fraction):
      frac = random_num_from_range(nim_fraction, max_fraction)
      n_atoms = round(frac * G.num_vertices)
      atoms = {random vertex from G}
      while |atoms| < n_atoms:
            edge = randomly choose edge having one vertex in atoms
                  and second not in atoms
            next_atom = choose vertex from edge not in atoms
            atoms += next_atom
      return inducted_subgraph(G, atoms)
```

Dávkovanie dát

Dávkovanie príkladov umožňuje škálovať tréning modelov hlbokého učenia na veľké množstvá dát. Dávkovanie, namiesto postupného spracovávania príkladov, združuje množiny príkladov do jednotnej reprezentácie, čo umožňuje ich efektívne spracovanie s využitím paralelizácie. Pri spracovaní obrazu alebo jazyka je jednotný formát dát dosiahnutý preškálovaním alebo doplnením príkladov tak, aby dosahovali rovnakú veľkosť. Takéto príklady sú potom združené pozdĺž novej dimenzie s veľkosťou dávky.

Keďže grafy sú jednou z najvšeobecnejších dátových štruktúr, ktoré vedia uchovať ľubovoľný počet vrcholov a hrán, vyššie spomenuté prístupy dávkovania sú buď neuskutočniteľné, alebo pri nich dochádza k prílišnému plytvaniu pamäte. Knižnica PyG implementuje dávkovanie grafov zjednotením grafov do jedného nesúvislého grafu, v ktorom pôvodné príklady tvoria komponenty súvislosti (Obr. 15).



Obr. 15: Dávkovanie grafov v knižnici PyG. A_1 až A_n sú matice susednosti jednotlivých grafov v dávke, X_1 až X_n ich vrcholové atribúty a Y_1 až Y_n ich značky.

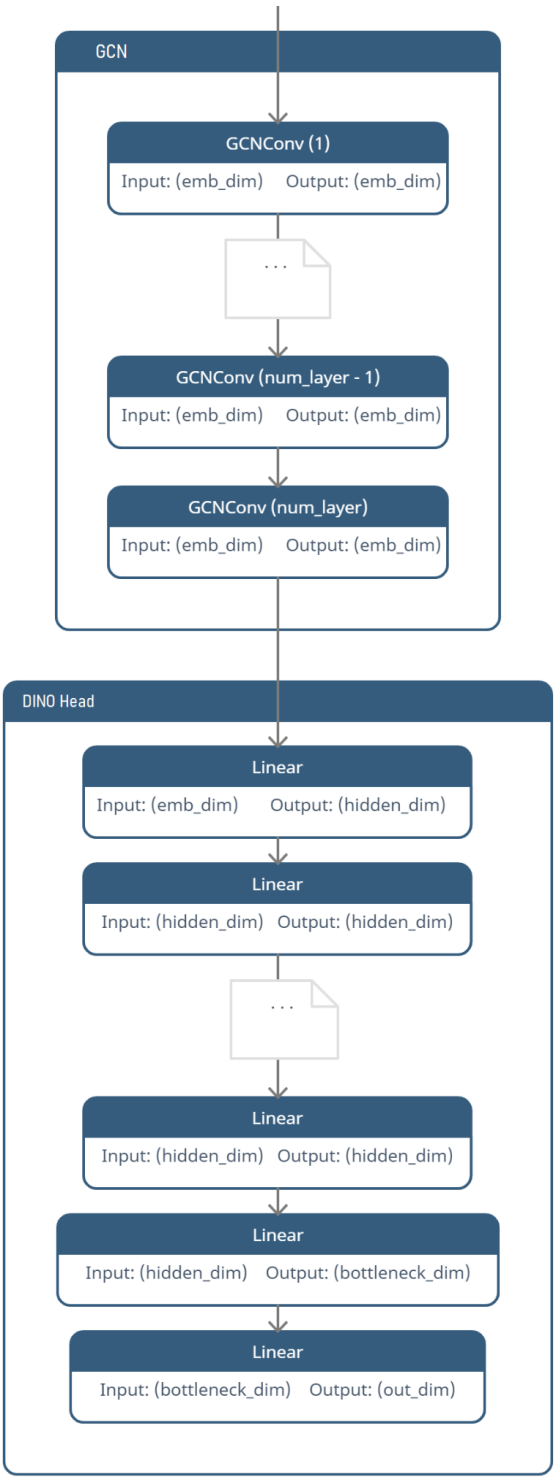
Použitie operátorov GNN založených na princípe message passing schémy zaručuje, že správy z rôznych príkladov v dávke sa navzájom nebudú miešať, teda dôjde k izolovanému spracovaniu všetkých grafov dávky efektívnym spôsobom s ohľadom na výpočtové a pamäťové prostriedky.

Návrh neurónovej siete

Neurónová sieť ktorú sme použili pozostáva z dvoch častí a to z GCN kostry siete a DINO hlavice (Obr. 16).

GCN kostru tvorí grafová konvolučná sieť pozostávajúca z niekoľkých grafových konvolučných vrstiev, implementovaných pomocou knižnice PyTorch Geometric (PyG), ktoré dedia od triedy MessagePassing. Počet týchto vrstiev udáva parameter `num_layers`. Vstupné a výstupné dimenzie sú identické a definuje ich parameter `emb_dim`.

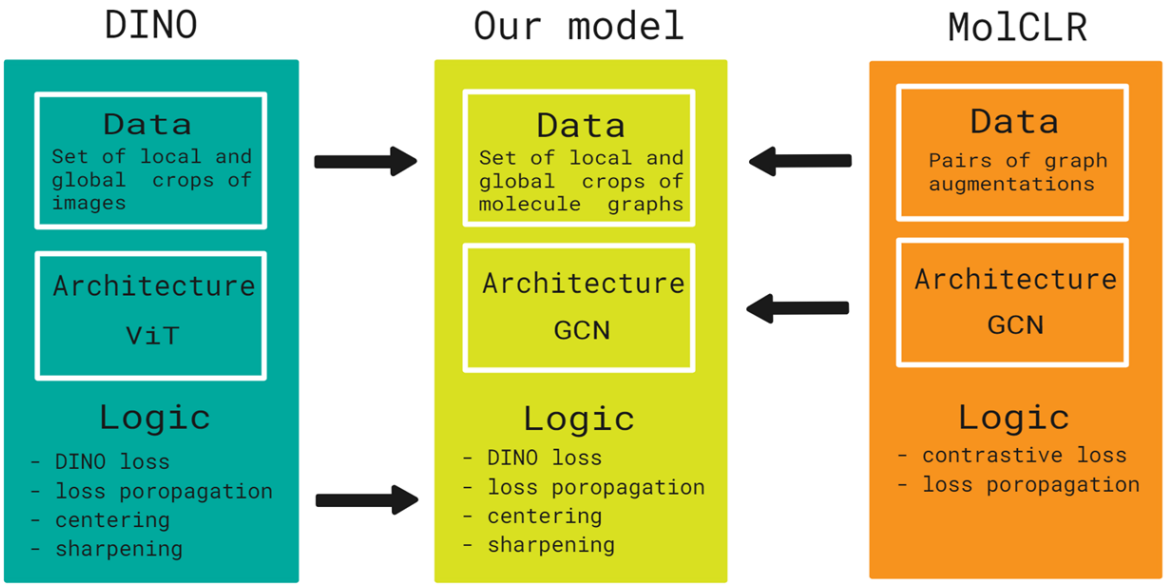
Tieto vrstvy nasleduje DINO hlava implementovaná ako viacvrstvový perceptrón (MLP), kde počet vrstiev určuje parameter `n_layers`. Prvá vrstva transformuje vstupy veľkosti `emb_dim` na vektory veľkosti `hidden_dim`. Ďalších `n_layers - 2` vrstiev rozmery dát nemení. Predposledná lineárna vrstva redukuje dáta na rozmer `bottleneck_dim`, ktoré následne posledná vrstva spätne rozširuje na dimenziu `out_dim`. V prípade, ak DINO hlava obsahuje len dve vrstvy, tak prvá vrstva hneď redukuje vstup na veľkosť `bottleneck_dim` a posledná vrstva ostáva nezmenená.



Obr. 16: Architektúra komponentov modelu. GCN sieť pozostáva z `num_layers` grafových konvolučných vrstiev. Tú nasleduje DINO hlava obsahujúca niekoľko ďalších lineárnych vrstiev postupne meniacich dimenziu dát, ktorú určujú parametre `emb_dim`, `hidden_dim`, `bottleneck_dim`, a `out_dim`. Počet vrstiev definuje parameter `n_layers`.

Návrh učiaceho algoritmu

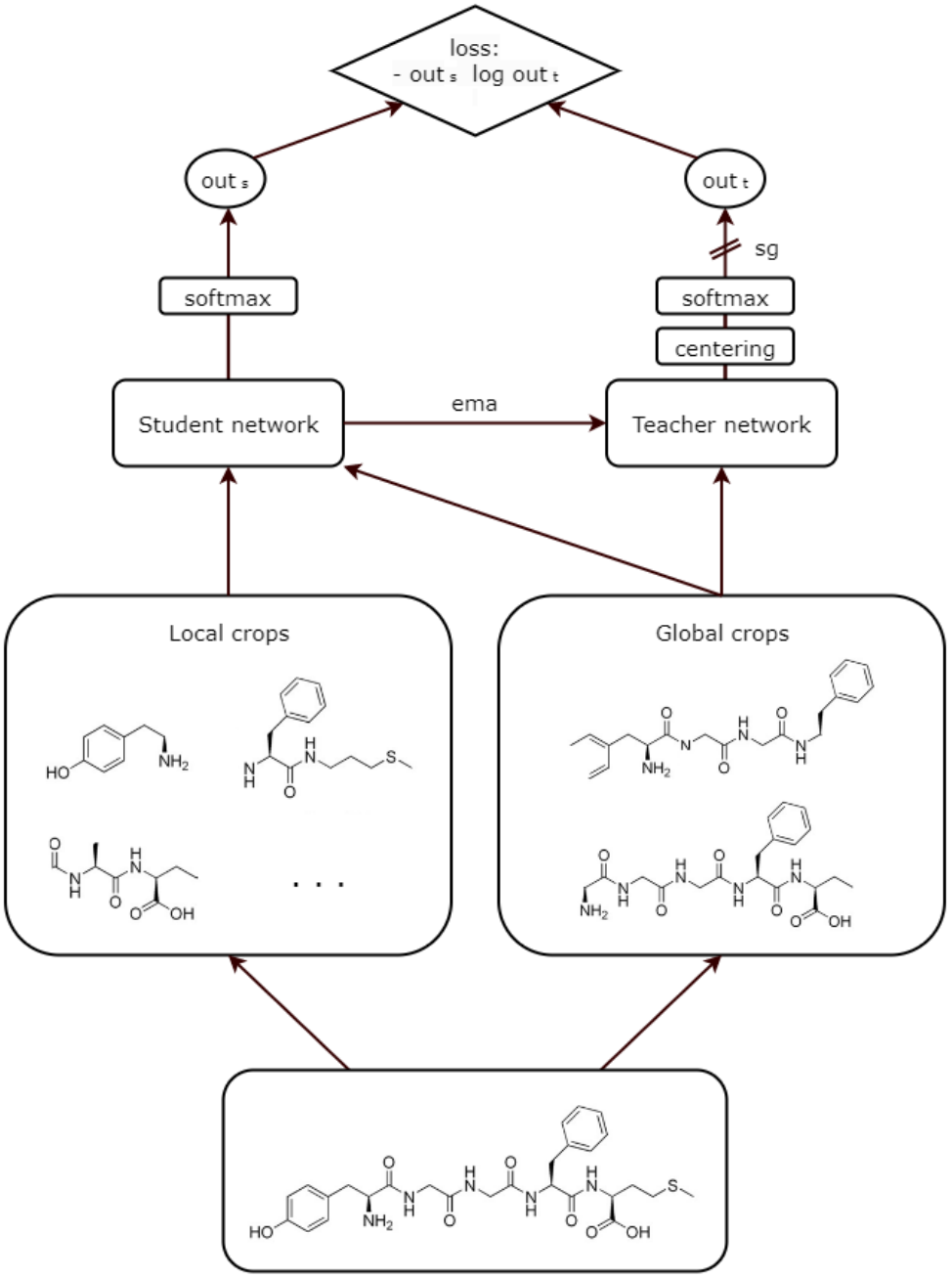
Pri návrhu algoritmu učenia sme využili princípy samokontrolovaného učenia DINO v kombinácii s metódami práce s grafmi, ktoré boli použité v algoritme MolCLR. Konkrétnejšie tieto princípy a ich pôvod zachytáva obrázok 17.



Obr. 17: Návrh modelu. Načítavanie a vnútornú reprezentáciu dát sme implementovali podľa vzoru kontrastívneho učenia MolCLR, ale dáta sú ďalej upravované a reorganizované podľa princípov samokontrolovaného učenia DINO. Jadro modelu tvorí grafová konvolučná sieť (GCN). Model je trénovaný na základe metód použitých v DINO, teda pomocou centeringu a sharpeningu, vlastnej chybovej funkcie a s využitím princípov destilácie neurónových sietí.

Dáta, v našom prípade grafy molekúl, sme načítavali po dávkach veľkosti `emb_dim` pomocou triedy Dataloader z knižnice PyG. Pre každý vstupný príklad sme vytvorili dva globálne výseky s rozmerom, ktorý bol určený rozsahom podielov `global_crops_scale` a menšie lokálne výseky veľkosti v rozmedzí `local_crops_scale` v počte `local_crops_number`. Trénovali sme dve identické neurónové siete s vyššie popísanou architektúrou: študentskú a učiteľskú (Obr. 18). Vstupom študentskej siete boli všetky výseky (lokálne aj globálne), no učiteľskej sieti boli počas tréningu poskytnuté len globálne výseky grafov molekúl. Výstupy učiteľskej siete boli modifikované pomocou centralizačnej metódy (centering), ktorá k výstupom pridáva zotrvačnosť predchádzajúcich predikcií.

Následne boli výsledky oboch sietí vyostrené pomocou nastavenia teploty softmax funkcie (sharpening). Teplota študentskej siete bola rovnaká pre všetky epochy, no teplota učiteľskej siete sa v priebehu tréningu dynamicky menila. Prvých `warmup_teacher_temp_epochs` epoch bola jej hodnota konštantná (`warmup_teacher_temp`) a následne dynamicky narastala na hodnotu `teacher_temp` pomocou kosínusového plánovača.



Obr. 18: Návrh aplikácie metódy DINO v chemickej doméne na spracovanie grafov molekúl.

Pri výpočte chyby sme použili funkciu krížovej entropie, kde hodnotu očakávaného výstupu poskytla učiteľská sieť a ako predikovaná hodnota bol použitý výstup študentskej siete. Celkovú hodnotu chyby pre výpočet nad jednou molekulou tvoril priemer hodnôt chyby vypočítaných medzi všetkými dvojicami, ktoré prislúchali výstupom navzájom rôznych vstupných výsekov danej molekuly.
Algoritmus spätného šírenia chyby a adaptácie váh študentskou sieťou určuje parameter `optimizer`, ktorý vyberá z dvoch možností. Jednou z nich je algoritmus stochastického poklesu gradientu (SGD) a ďalšou je modifikácia algoritmu Adam, algoritmus AdamW. K šíreniu chyby učiteľskou sieťou nedochádza, keďže je na ňu aplikovaný operátor stop gradient, ale učenie tejto siete je implementované pomocou EMA metódy (exponential moving average):

$$W_{teacher} = m * W_{teacher} + (1 - m) * W_{student},$$

kde $W_{teacher}$ a $W_{student}$ sú váhy učiteľskej a študentskej siete a m je pomer určený paramterom `momentum_teacher`.

Hyperparametre

Hyperparametre algoritmu strojového učenia sú parametre, ktorých hodnoty kontrolujú učiaci proces a determinujú hodnoty váh modelu na konci behu učiaceho algoritmu. Tieto parametre sú definované pri dizajnovaní modelu a ich hodnoty ostávajú nemenné počas učenia modelu.

Pri návrhu nášho algoritmu sme identifikovali viacero hyperparametrov, ktoré sme kvôli prehľadnosti rozdelili do niekoľkých skupín a sú vysvetlené v nasledujúcich tabuľkách. Všeobecné hyperparametre (Tab. 3), hyperparametre pre spracovanie dát (Tab. 4), hyperparametre určujúce architektúru kostry siete a projekčnej hlavy (Tab. 5).

Tab. 3: Všeobecné hyperparametre, ktoré definujú dĺžku tréningu, veľkosť dávok, vývoj učiaceho pomeru, inicializujú regularizačné metódy a popisujú ďalšie iniciálne hodnoty učiaceho algoritmu.

Hyperparameter	Funkcia hyperparametra
epochs	počet epoch
batch_size	veľkosť dávky
optimizer	algoritmus spätného šírenia chyby
clip_grad	parameter metódy gradient clipping, ktorá zabraňuje gradientu aby nadobudol príliš veľké hodnoty
freeze_last_lazer	počet epoch počas ktorých sú parametre výstupnej vrstvy nemenné
use_fp16_precision	boolean príznak určujúci presnosť desatinných čísel počas tréningu
lr	učiaci pomer na konci lineárneho zahrievania
min_lr	učiaci pomer na konci algoritmu
warmup_epochs	počet epoch lineárneho zahrievania učiaceho pomeru
weight_decay	počiatočná hodnota parametra regularizačnej metódy
weight_decay_end	koncová hodnota parametra regularizačnej metódy
momentum_teacher	parameter určujúci moment EMA algoritmu
student_temp	teplota softmax funkcie študentskej siete
warmup_teacher_temp	počiatočná hodnota teploty softmax funkcie učiteľskej siete, ktorá je počas prvých epoch konštantná
warmup_teacher_temp_epochs	počet epoch, počas ktorých je teplota softmax funkcie učiteľskej siete konštantná
teacher_temp	teplota softmax funkcie učiteľskej siete na konci tréningu

Tab. 4: Prehľad hyperparametrov pre prácu s dátami, ktoré určujú veľkosti a počty výsekov a načítavanie dát.

Hyperparameter	Funkcia hyperparametra
global_crops_scale	interval pomerov určujúcich veľkosť globálnych výsekov
local_crops_scale	interval pomerov určujúcich veľkosť lokálnych výsekov
local_crops_number	počet lokálnych výsekov

Hyperparameter	Funkcia hyperparametra
num_workers	počet procesov určených na načítavanie dát
valid_size	pomer validačných dát

Tab. 5: Hyperparamtre, ktoré sa podieľajú na architektúre neurónových sietí.

Hyperparameter	Funkcia hyperparametra
n_GCN_layers	počet grafových konvolučných vrstiev
emb_dim	veľkosť dimenzie vstupov a výstupov GCN
drop_ratio	parameter metódy dropout, ktorá bráni preučeniu
pool	poolingová operácia v GCN
n_dino_layers	počet vrstiev v projekčnej hlave
hidden_dim	počet neurónov v skrytej vrstve projekčnej hlavy
bottleneck_dim	počet neurónov vo vrstve redukujúcej tok dát
out_dim	počet neurónov vo výstupnej vrstve
use_bn	boolean príznak určujúci, normalizovanie naprieč dávkou
norm_last_lazer	boolean príznak určujúci, či váhy poslednej vrstvy majú byť normalizované

Zdroje:

1. Zhenqin Wu and Bharath Ramsundar and Evan N. Feinberg and Joseph Gomes and Caleb Geniesse and Aneesh S. Pappu and Karl Leswing and Vijay S. Pande, . "MoleculeNet: A Benchmark for Molecular Machine Learning". CoRR abs/1703.00564. (2017).
2. DeepChem - https://deepchem.readthedocs.io/en/latest/index.html
3. Mathilde Caron and Hugo Touvron and Ishan Misra and Hervé Jégou and Julien Mairal and Piotr Bojanowski and Armand Joulin, . "Emerging Properties in Self-Supervised Vision Transformers". CoRR abs/2104.14294. (2021).
4. Mahmoud Assran and Mathilde Caron and Ishan Misra and Piotr Bojanowski and Armand Joulin and Nicolas Ballas and Michael G. Rabbat, . "Semi-Supervised Learning of Visual Features by Non-Parametrically Predicting View Assignments with Support Samples". CoRR abs/2104.13963. (2021).
5. Yuyang Wang and Jianren Wang and Zhonglin Cao and Amir Barati Farimani, . "MolCLR: Molecular Contrastive Learning of Representations via Graph Neural Networks". CoRR abs/2102.10056. (2021).
6. Seyone Chithrananda and Gabriel Grand and Bharath Ramsundar, . "ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction". CoRR abs/2010.09885. (2020).
7. Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In CVPR, (2020).
8. Jean-Bastien Grill, Florian Strub, Florent Altch´e, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, R´emi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In NeurIPS, (2020).
9. Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. preprint arXiv:2002.05709, (2020).
10. Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan and Ilya Sutskever. Generative Pretraining from Pixels. In PMLR, (2020)

11. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. (2017).
12. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
13. On DINO, Self-Distillation with no labels https://towardsdatascience.com/on-dino-self-distillation-with-no-labels-c29e9365e382
14. BERT Technology introduced https://towardsdatascience.com/bert-technology-introduced-in-3-minutes-2c2f9968268c
15. BERT, RoBERTa, DistilBERT, XLNet https://towardsdatascience.com/bert-roberta-distilbert-xlnet-which-one-to-use-3d5ab82ba5f8
16. Han Altae-Tran, Bharath Ramsundar, Aneesh S. Pappu, and Vijay Pande. Low Data Drug Discovery with One-shot Learning. 2016
17. David Duvenaud† , Dougal Maclaurin† , Jorge Aguilera-Iparraguirre Rafael Gomez-Bombarelli, Timothy Hirzel, Al ´ an Aspuru-Guzik, Ryan P. Adams. Convolutional Networks on Graphs for Learning Molecular Fingerprints 2015