



Kandidatutkielma

Tietojenkäsittelytieteen kandiohjelma

Koneoppiminen lääkkeiden kehityksessä

Heikki Pulli

7.12.2021

MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
HELSINGIN YLIOPISTO

Ohjaaja(t)

Prof. D.U. Mind, Dr. O. Why

Tarkastaja(t)

Prof. D.U. Mind, Dr. O. Why

Yhteystiedot

PL 68 (Pietari Kalmin katu 5)
00014 Helsingin yliopisto

Sähköpostiosoite: info@cs.helsinki.fi

URL: <http://www.cs.helsinki.fi/>

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty		Koulutusohjelma — Utbildningsprogram — Study programme	
Matemaattis-luonnontieteellinen tiedekunta		Tietojenkäsittelytieteen kandiohjelma	
Tekijä — Författare — Author			
Heikki Pulli			
Työn nimi — Arbetets titel — Title			
Koneoppiminen lääkkeiden kehityksessä			
Ohjaajat — Handledare — Supervisors			
Prof. D.U. Mind, Dr. O. Why			
Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages	
Kandidutkielma	7.12.2021	26 sivua	
Tiivistelmä — Referat — Abstract			
<p>Tämä dokumentti on tarkoitettu Helsingin yliopiston tietojenkäsittelytieteen osaston opinnäytteiden ja harjoitustöiden ulkoasun ohjeeksi ja mallipohjaksi. Ohje soveltuu kandidutkielmiin, ohjelmistotuotantoprojekteihin, seminaareihin ja maisterintutkielmiin. Tämän ohjeen lisäksi on seurattava niitä ohjeita, jotka opastavat valitsemaan kuhunkin osioon tieteellisesti kiinnostavaa, syvällisesti pohdittua sisältöä.</p> <p>Työn aihe luokitellaan ACM Computing Classification System (CCS) mukaisesti, ks. https://www.acm.org/about-acm/class, käyttäen komentoa <code>\classification{}</code>. Käytä muutamaa termipolkua (1–3), jotka alkavat juuritermistä ja joissa polun tarkentuvat luokat erotetaan toisistaan oikealle osoittavalla nuolella.</p> <p>ACM Computing Classification System (CCS) General and reference → Document types → Surveys and overviews Applied computing → Document management and text processing → Document management → Text editing</p>			
Avainsanat — Nyckelord — Keywords			
ulkoasu, tiivistelmä, lähdeluettelo			
Säilytyspaikka — Förvaringsställe — Where deposited			
Helsingin yliopiston kirjasto			
Muita tietoja — övriga uppgifter — Additional information			
Ohjelmistojärjestelmien erikoistumislinja			

Sisällys

1	Johdanto	1
2	Kemiallinen data	3
2.1	Yhdisteiden kuvaaminen merkkijonoilla	3
2.2	Ominaisuusvektorit	4
3	Uusien lääkkeiden löytäminen	5
3.1	Lääkkeiden löytäminen	5
3.2	Virtuaalinen seulonta	5
3.3	Kehitettyjä koneoppimismalleja	6
3.3.1	CMG	6
3.3.2	SPN	8
4	Uusien lääkkeiden syntetisointi	10
4.1	Yhdisteiden syntetisointi	10
4.2	Lääkkeen retrosyntetisoinnin haastavuus	11
4.3	Kehitettyjä apuvälineitä	12
4.3.1	3N-MCTS	12
4.3.2	Heurustiikka avusteiset neuroverkot	18
5	Lääkkeiden kehitys tulevaisuudessa	22
6	Yhteenveto	23
	Lähteet	25

1 Johdanto

Uusien lääkkeiden kehittäminen on pitkä ja kallis prosessi. Tähän kuuluu useita vaiheita ja eri vaiheet vievät eri määrän rahaa ja aikaa. Vaiheet ovat sairauden aiheuttajan tunnistaminen, aiheuttajaan vaikuttavan lääkkeen tunnistaminen, lääkkeen optimointi, lääkkeen ominaisuuksien analysointi ja kliiniset testit. Näiden vaiheiden jälkeen lääke joko hyväksytään myyntiin tai sitä joudutaan edelleen jatkokehittämään. Kaikki vaiheet yhdessä ovat tavallisesti vieneet 10–12 vuotta ja kaikkien vaiheiden jälkeen työn hinnaksi on voinut muodostua 1–3 miljardia dollaria [5].

Vaiheiden pitkän keston ja korkean hinnan takia tutkijat ja lääkealan yritykset ovatkin alkaneet tutkia mahdollisia keinoja, jotka nopeuttaisivat lääkkeiden tutkimista tai lasisivat lääkekehityksen prosessiin kuluva hintaa. Koneoppimismallit ovat nousseet houkuttelevaksi vaihtoehdoksi ratkaisemaan näitä ongelmia. Yritykset ovatkin alkaneet selvittää, kuinka eri koneoppimismalleja voidaan hyödyntää nopeuttamaan lääketutkimuksen suurimpia pullonkauloja [5].

Viimeisen kymmenen vuoden aikana saatavilla olevan laadukkaan datan määrä on kasvanut merkittävästi ja samalla on kehitetty uusia tehokkaampia koneoppimismalleja, joita voidaan hyödyntää lääketutkimuksessa. Koneoppimismallien avulla voidaan esimerkiksi karsia kaikista harkinnasta olevista lääkkeistä lupaavimmat kandidaatit, joilla on mahdollisuus päästä testeistä läpi tuotantoon. Koneoppimismalleja voidaan myös hyödyntää aivan uusien lääkkeeksi sopivien yhdisteiden etsinnässä. Tähän tarkoitukseen kehitetyt koneoppimismallit kykenevät etsimään lääkkeitä, joilla on halutut lääkkeelliset ja fysikaaliset ominaisuudet [12]. Kehitetyt mallit ovatkin näyttäneet, että koneoppimismallit ovat tehokkaita työkaluja, joita voidaan hyödyntää kaikissa lääketutkimuksen eri vaiheissa.

Tässä tekstissä paneudutaan syvemmin koneoppimismalleihin, joita käytetään uusien lääkkeiden tunnistamiseen ja näiden tunnistettujen lääkkeiden syntetisoinnin suunnitteluun.

Ensimmäisessä kappaleessa tutustutaan eri datatyyppeihin, joita käytetään koneoppimismallien kouluttamisessa. Tämä on lyhyt kappale, jossa käydään vain ne datatyypit läpi, joita myöhemmin selitettävät koneoppimismallit käyttävät. Näiden ymmärtäminen auttaa hahmottamaan myöhemmin tekstissä selitettävien eri mallien toimintaperiaatteita.

Toisessa kappaleessa käydään läpi, kuinka uusia lääkkeitä löydetään. Kappaleessa kerro-

taan virtualisesta seulonnasta (Virtual screening), mitä sillä tarkoitetaan ja kuinka sitä hyödynnetään. Lisäksi toisessa kappaleessa käydään läpi yhden koneoppimismallin toimintaa, joka on koulutettu löytämään uusia yhdisteitä. Tämä malli mahdollistaa nopean uusien lääkkeiden löytämisen, joita näin ollen päästään nopeasti jatkotutkimaan.

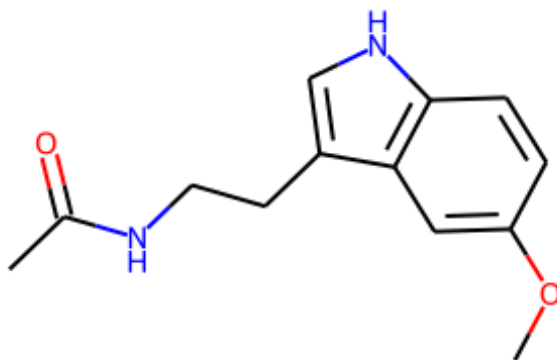
Kolmannessa kappaleessa käsitellään yhdisteiden syntetisointia. Kappaleessa kerrotaan, mikä tekee yhdisteiden syntetisoinnista hankalaa ja mitä ratkaisuja tähän on kehitetty. Kappale myös kertoo retrosyntetisoinnista ja CASP -menetelmistä (**Computer Aided Synthesis Planning**) ja kuinka nämä liittyvät toisiinsa. Lisäksi kappaleessa käydään läpi kahden eri koneoppimismallin toimintaa, jotka on kehitetty auttamaan eri yhdisteiden syntetisoinnissa.

Neljäs kappale käsittelee, kuinka koneoppimista tullaan käyttämään tulevaisuuden lääkekehityksessä ja tutkimisessa. Kappale kertoo, kuinka koneoppiminen tulee enenemissä määrin muuttamaan lääkekehityksen prosessia.

2 Kemiaallinen data

Tässä kappaleessa käydään pintapuolisesti läpi, minkälaista dataa käytetään mallien kouluttamiseen. Kappaleessa keskitytään datatyyppeihin, joita käytetään malleissa, joiden toimintaa tullaan avaamaan tämän tekstin muissa osissa. Mallit, joiden toimintaa avataan, käyttävät datatyyppejä, joilla kuvataan pienyhdisteitä. Tämän takia kappaleessa keskitytään pieniäyhdisteitä kuvaaviin datatyyppeihin. Kappaleessa käydään läpi SMILES ja SMARTS datatyypit, mihin niitä käytetään ja kuinka ne eroavat toisistaan. Lisäksi kappaleessa kerrotaan ominaisuusvektoreista, kuinka niitä käytetään ja missä tilanteissa.

2.1 Yhdisteiden kuvaaminen merkkijonoilla



CC(=O)NCCC1=CNC2=C1C=C(C=C2)OC

Kuva 2.1: Melatoniini yhdiste, kuinka se kuvataan graaffina ja SMILES -merkkijonona. Kuva on tuotettu hyödyntämällä RDKit -kirjastoa.

[10]

Tavallinen tapa kuvata yhdisteitä on kuvata ne yhdellä rivillä merkkijonoina. SMILES (Simplified Molecular Input Line Entry System) on David Weiningerin vuonna 1988 kehittämä tapa kuvata molekyyliä ja yksinkertaisia yhdisteitä [14]. Se kuvaa yhdisteen yhdellä rivillä käyttäen ASCII -merkistöä. Tämän takia se on tietokoneen nopea lukea ja

kemistin helppo ymmärtää.

SMARTS on SMILES -standardista jatkokehitetty tapa kuvata yhdisteitä. Se mahdollistaa yhdisteiden sisäisten rakenteiden määrittämisen. SMARTS on yläluokka SMILES standardista, jonka takia kaikki SMILES -merkkijonoiksi kelpaavat merkkijonot ovat myös kelpoja SMARTS -merkkijonoja. SMARTS ja SMILES standardit mahdollistavat myös kemiallisten reaktioiden kuvaamisen ASCII -merkistöllä yhdellä rivillä.

2.2 Ominaisuusvektorit

Yksi tässä tekstissä käsiteltävistä koneoppimismalleista käyttää ominaisuusvektoreita hyödykseen. Ominaisuusvektorit sisältävät tietoa yhdisteen tai molekyylin lääkkeellisistä ominaisuuksista. Ominaisuusvektorit voivat kuvata mitä tahansa yhdisteen tai molekyylin ominaisuutta ja ne voidaan valita tilanteen mukaan. Kyseisessä mallissa ominaisuusvektorit kuvaavat molekyylin PlogP, QED ja DRD2 arvoja. PlogP (Penalized logP) kuvaa aineen rasvaliukoisuutta, QED kuvaa, kuinka paljon tietty yhdiste on lääkkeenkaltainen ja DRD2 kuvaa, kuinka paljon molekyyli reagoi dopamiini 2 receptoria vastaan [2].

3 Uusien lääkkeiden löytäminen

Tässä kappaleessa kerrotaan, mitä asioita liittyy lääkkeiden löytämiseen. Lisäksi kappaleessa kerrotaan, mitä on virtuaalinen seulonta ja kuinka sitä hyödynnetään lääkkeiden löytämisessä. Kappaleessa myös käydään läpi yhden koneoppimismallin toimintaperiaate. Kyseinen koneoppimismalli luo uusia lääkkeitä perustuen ominaisuusvektoreihin.

3.1 Lääkkeiden löytäminen

Yksi ensimmäisistä lääkekehityksen osa-alueista on uusien lääkkeiksi kelpaavien yhdisteiden löytäminen joko uusiin tai jo tunnettuihin tauteihin [5]. Lääkeyhdisteiden löytäminen on kuitenkin ollut tavallisesti hidas prosessi ja uuden toimivan yhdisteen löytäminen on kestänyt useita vuosia.

Viimeisimmän kymmenen vuoden aikana saatavilla olevan datan määrä on kasvannut merkittävästi kehitettyjen tietopankkien takia [5]. Näitä ovat esimerkiksi PubChem ja ChEMBL. Lisääntynyt datan määrä on täten mahdollistanut lääkkeiden löytämisen nopeuttamisen koneoppimismallien avulla.

Popovan et al. tutkimusryhmä on kehittänyt mallin, joka ehdottaa uutta yhdistettä perustuen mallin syötteenä saamaan ominaisuusvektoriin [8].

3.2 Virtuaalinen seulonta

Kemiallisten yhdisteiden avaruudella tarkoitetaan kaikkien uniikkien yhdisteiden lukumäärää. On arvioitu, että erilaisia kemiallisia yhdisteitä, jotka voivat esiintyä huoneenlämmössä nesteessä, voi olla välillä $10^{18} - 10^{180}$ [11]. Yhdisteet, jotka täyttävät lääkkeitä vaaditut kriteerit, on puolestaan arvioitu olevan noin 10^{60} [11]. Erillaisten yhdisteiden suuri lukumäärä itsessään esittää tarpeen tehokkaille algoritmeille ja menetelmille, jotka auttavat karsimaan tästä suuresta määrästä kemiallisia yhdisteitä vain lupaavimmat.

Virtuaaliseulonta (Virtual Screening, VS) on suosittu lähestymistapa uusien lääkkeiden löytämiseksi. Virtuaaliseulonta käsittää joukon menetelmiä, joissa tietokoneita hyväksi käyttäen karsitaan yhdisteiden avaruudesta vain tietyt kriteerit täyttävät yhdisteet [11].

Yhdisteiden karsimiseen käytetään ohjelmistoja, joissa toteutetut algoritmit suorittavat seulontaa annetuilla parametreilla. Koneoppimismenetelmiä voidaan myös käyttää virtuaalisessa seulonnassa. Virtuaaliseulonnan avulla löydettyjä yhdisteitä voidaan sitten jatkokotutkia ja edelleen kehittää lääkkeiksi.

3.3 Kehitettyjä koneoppimismalleja

Käydään seuraavaksi läpi yhden toteutetun koneoppimismallin toimintaperiaate. Kyseinen malli etsii uusia lääkkeitä käyttämällä hyödyksi sille annettua prototyyppiyhdistettä, prototyypin kemiallisia ominaisuuksia ja uudelta lääkkeeltä haluttuja kemiallisia ominaisuuksia. Kemialliset ominaisuudet on koodattu ominaisuusvektoreihin.

3.3.1 CMG

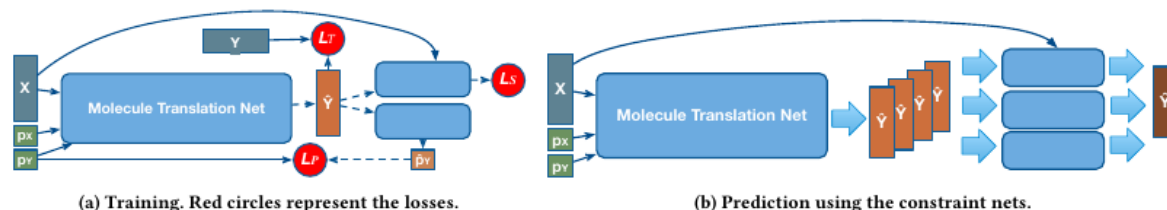
CMG (Controlled molecule generator) on koneoppimismalli, joka etsii uusia molekyyliä, jotka perustuvat syötteenä annettuun molekyyliin ja joilla on ennalta määritellyt halutut ominaisuudet [10]. Se eroaa aikaisemmin kehitetyistä malleista siten, että se pystyy optimoimaan annetun molekyylin useampaa ominaisuutta samanaikaisesti. Aikaisemmin kehitetyt mallit ovat pystyneet optimoimaan vain yhtä molekyylin ominaisuutta.

CMG:n kehittäjät lähestyvät molekyylin ominaisuuksien optimointiongelmia merkkijonojen käännös -ja luontiongelmiana. CMG:lle opetetaan, kuinka syötteenä annettu yhdistemerkkijono käännetään yhdistemerkkijonoksi, jolla on lähimpänä haluttuja ominaisuuksia olevat ominaisuudet. CMG tulkitsee ensin annetut merkkijonot hyödyntäen DN:ää (deep network) jonka jälkeen se luo uusia molekyyliyhdisteitä hyödyntäen tätä tulkintaa ja haluttua ominaisuusvektoria. Koska ominaisuudet annetaan vektorina, niin CMG pystyy optimoimaan useampaa ominaisuutta samanaikaisesti. CMG lisäksi hyödyntää ennalta koulutettuja rajoiteverkkoja (constraint network, CN), jolloin vältetään luomasta mahdollottomia yhdisteitä. CMG -malli käyttää näitä verkkoja hyödyksi käyttämällä muokattua Beam Search -algoritmia.

CMG -malli muodostuu kolmesta verkosta, joista kaksi toimivat rajoiteverkkoina. Ensimmäinen verkko on MTN (**Molecule Translation Network**) ja rajoiteverkot ovat PPN (**Property Prediction Network**) ja SPN (**Similarity Prediction Network**).

Data, jolla mallit koulutetaan, on peräisin ZINC ja DRD2 dataseiteistä. (Kuka hallinnoi?) Data sisältää kaiken kaikkiaan 257 565 molekyyliä, joista luodaan pareja. Parit muodos-

tetaan siten, että kahden molekyylin välinen samanlaisuysluku on yli 0.4. (Miten lasketaan?) Näitä pareja (X, Y) muodostetaan kaiken kaikkiaan 10 827 615 kappaletta. Lisäksi kaikille molekyyleille lasketaan niiden eri kemiallisten ominaisuuksien arvot, jotka ovat PlogP, QED ja DRD2. Nämä kuvaavat eri ominaisuuksia, joita malli pyrkii optimoimaan halutulla tavalla.



Kuva 3.1: CMG:n toiminta periaate kuvattuna

[10]

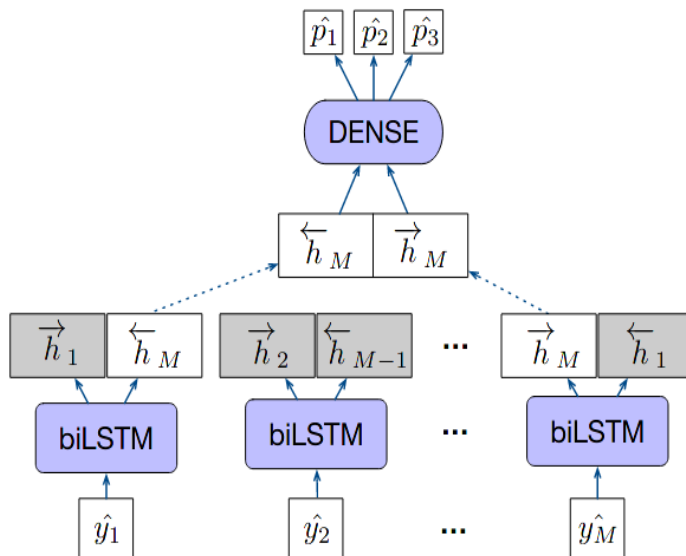
MTN

MTN -verkko perustuu aikaisemmin kehitettyyn The Transformer -malliin [13]. MTN eroaa Transformer -mallista kahdella tavalla. Toisin kuin Transformer -malli, joka käsittelee sanoja ja niistä muodostettuja lauseita, MTN -verkko käsittelee yksittäisiä merkkejä ja niistä muodostettuja molekyylejä. Lisäksi MTN:än piilotettuun kerrokseen on lisätty tietoa kemiallisisista ominaisuuksista.

PPN

PPN -verkko ottaa syötteenä ennustetun molekyylin merkkijonon (y_i) . Tämä merkkijono muunnetaan piilotetuiksi vektoreiksi hyödyntäen Bidirectional Long short-term memory (biLSTM) -kerrosta. BiLSTM on rakenteeltaan samanlainen kuin LSTM taso, mutta koska se käsittelee syötteen sekä alku-loppu suunnassa että loppu-alku suunnassa se oppii tietyn merkin konteksin molemmissa suunnissa. BiLSTM -kerros muodostaa vektoreita oikealta vasemmalle ja vasemmalta oikealle suunnassa, ja näistä vektoreista valitaan molempien suuntien viimeiset vektorit. Nämä vektorit yhdistetään ja yhdistevektori syötetään täysin yhdistetylle neuroverkolle. Tämä verkko sisältää kaksi piilotettua tasoa. Verkko antaa tuloksena kyseisen yhdisteen ominaisuuksien ennusteen.

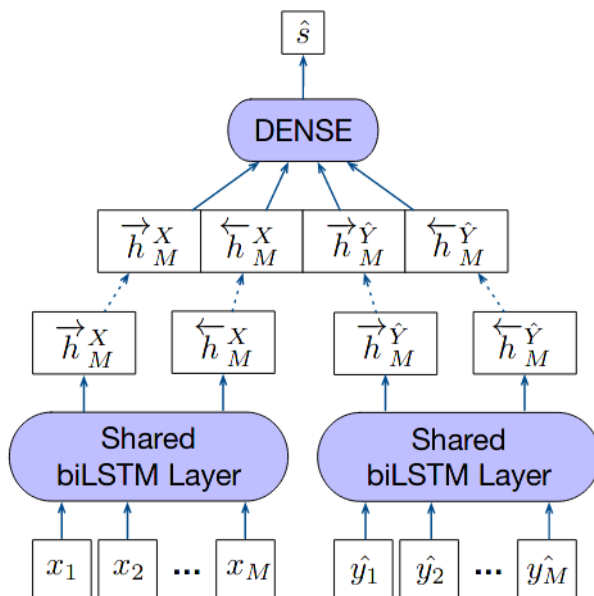
PPN koulutetaan käyttäen 257 565 eri molekyyliä. nämä molekyylit jaetaan satunna-raisella valinnalla koulutus -ja testiryhmiin suhteella 8:2.



Kuva 3.2: CMG -mallin PPN -neuroverkon rakenne

[10]

3.3.2 SPN



Kuva 3.3: CMG -mallin SPN -neuroverkon rakenne

[10]

SPN ottaa syötteenä ennustetun molekyylin merkkijonon (y_i) ja alkuperäisen molekyylin merkkijonon (x_i). Nämä merkkijonot syötetään kahdelle eri biLSTM -tasolle, toinen käsit-

telee alkuperäisen molekyylin merkkijonot ja toinen ennustetun molekyylin merkkijonot. Nämä biLSTM -kerrokset toimivat samalla periaatteella kuin PPN:ässä ja palauttavat näin ollen neljä eri vektoria, kaksi ennustetulta merkkijonolta ja kaksi alkuperäiseltä merkkijonolta. Nämä vektorit yhdistetään ja tämä yhdistevektori annetaan täysin yhdistetylle verkolle, joka sisältää kaksi piilotettua tasoa.

SPN kouluttamiseen käytetään osajoukkoa 10 827 615 eri parista. Tästä määrästä valitaan kymmenen prosentin osajoukko, joka on 997 773 molekyylin suuruinen. Tämä joukko jaetaan samalla tavalla koulutus ja testi ryhmiin kuin PPN:ässä.

4 Uusien lääkkeiden syntetisointi

Tässä kappaleessa käydään läpi, miten lääkkeiden syntetisoinnin suunnittelua voidaan tehostaa koneoppimismallien avulla. Kappaleessa keskitytään syntetisointiin retrosyntetisoinnin avulla. Kappaleessa kerrotaan, mikä tekee yhdisteen retrosyntetisoinnista vaikeaa ja kuinka sitä voidaan tehostaa eri koneoppimismallien avulla. Kappaleessa käydään läpi kahden koneoppimismallin toimintaa, jotka suorittavat halutun yhdisteen retrosyntetisointia.

4.1 Yhdisteiden syntetisointi

Yhdisteen syntetisoinnin suunnittelulla tarkoitetaan prosessia, jossa määritellään, kuinka haluttu yhdiste voidaan tuottaa synteettisesti saatavilla olevista lähtöaineista [4]. Retrosynteesianalyysillä tarkoitetaan puolestaan menetelmää, jonka avulla löydetään halutun yhdisteen tuottamiseen tarvittavat lähtöaineet. Retrosynteesi toimii siis toiseen suuntaan kuin syntetisointi. Retrosynteesissä yhdiste pilkotaan rekursiivisesti pienempiin lähtöaineisiin kunnes jäljellä on vain saatavilla olevia lähtöaineita.

Tavallisesti yhdisteen retrosyntetisointi on vaatinut suorittavalta kemistiltä usean vuoden kokemusta ja tietoa saatavilla olevista lähtöaineista ja eri reaktioista. Tätä on pyritty automatisoimaan eri CASP -menetelmien avulla (Computer-Aided Synthesis Planning). Ensimmäiset CASP -menetelmät perustuivat heuristisiin algoritmeihin, joissa kemistit käsin koodasivat, miten eri lähtöaineet reagoivat keskenään ja mikä on reaktion lopputuote. Tämä on kuitenkin osoittautunut toivottomaksi yritykseksi massiivisen data määrän takia.

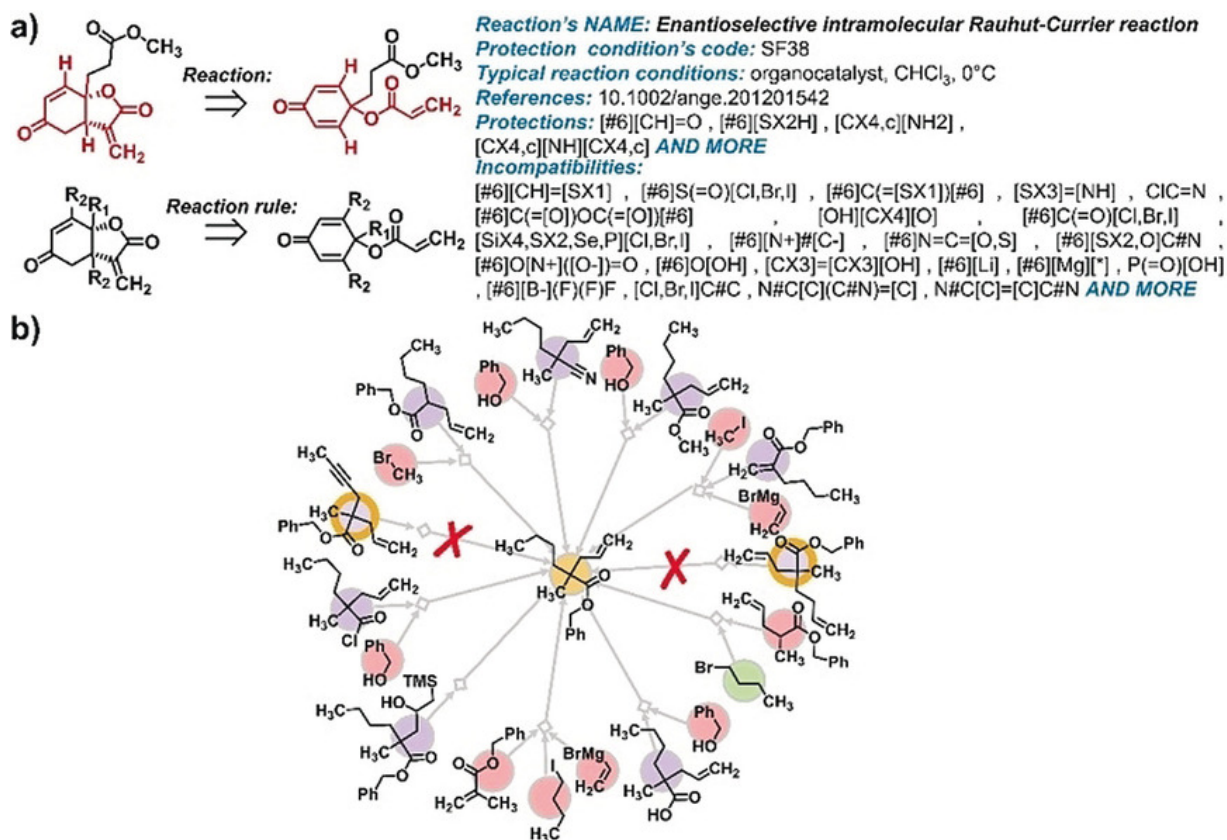
Kehitys koneoppimismenetelmissä on kuitenkin tarjonnut uuden lähestymistavan CASP -menetelmien kehitykseen. Sen sijaan, että kemistit kehittäisivät heuristisia malleja, uudet koneoppimismallit koulutetaan saatavilla olevan datan avulla. Tämä on todettu merkittävästi toteutettavammaksi lähestymistavaksi.

Koneoppimismallien käyttö ja koulutus ei ole kuitenkaan täysin ongelmaton lähestymistapa. Ongelmaan liittyvää dataa ei välttämättä ole saatavilla ja datan hankkiminen voi olla kallista. Tätä varten on kehitetty tietopankkeja, jotka sisältävät suuria määriä dataa tietystä aiheesta, esim. Reaxys kemiallisista reaktioista.

4.2 Lääkkeen retrosyntetisoinnin haastavuus

Retrosyntetisoinnin tekee hankalaksi se, että yhdiste voidaan muodostaa sadoilla tai tuhansilla eri tavoilla. Tämä ongelma toistuu rekursiivisesti, kun yhdiste pilkotaan yhdisteisiin, jotka keskenään reagoidessa muodostavat alkuperäisen yhdisteen. Pienille ja yksinkertaisille yhdisteille tämä vaihtoehtoavaruus on pienempi, mutta yhdisteen koon kasvaessa eri tapojen määrä muodostaa haluttu yhdiste kasvaa eksponentiaalisesti.

Tästä johtuen prosessia yleistäville koneoppimismalleille on suuri.



Kuva 4.1: (a) esimerkki reaktiosäännöstä, (b) esimerkki mahdollisista reaktioista, kuinka voidaan luoda haluttu yhdiste (keskellä). Kuvaa on yksinkertaistettu ja se sisältää vain yksitoista mahdollista reaktiota, jotka tuottavat halutun yhdisteen. Värikoodaukset tarkoittavat: Keltainen — haluttu yhdiste, Punainen — saatavilla oleva yhdiste, Vihreä — kirjallisuudesta tunnettu yhdiste, liila — tuntematon yhdiste

4.3 Kehitettyjä apuvälineitä

Seuraavissa kappaleissa avataan kahden kehitetyn koneoppimismallin toimintaa. Nämä mallit ovat toteutettu suorittaman retrosynteesiä, mutta niiden toimintaperiaate perustuu eri lähestymistapaan. 3N-MCTS -mallin toimintaperiaate perustuu täysin oppimiseen datasta, kun taas ICHO -malli hyödyntää myös ammattikemistien luomia heuristisiasäätöjä.

4.3.1 3N-MCTS

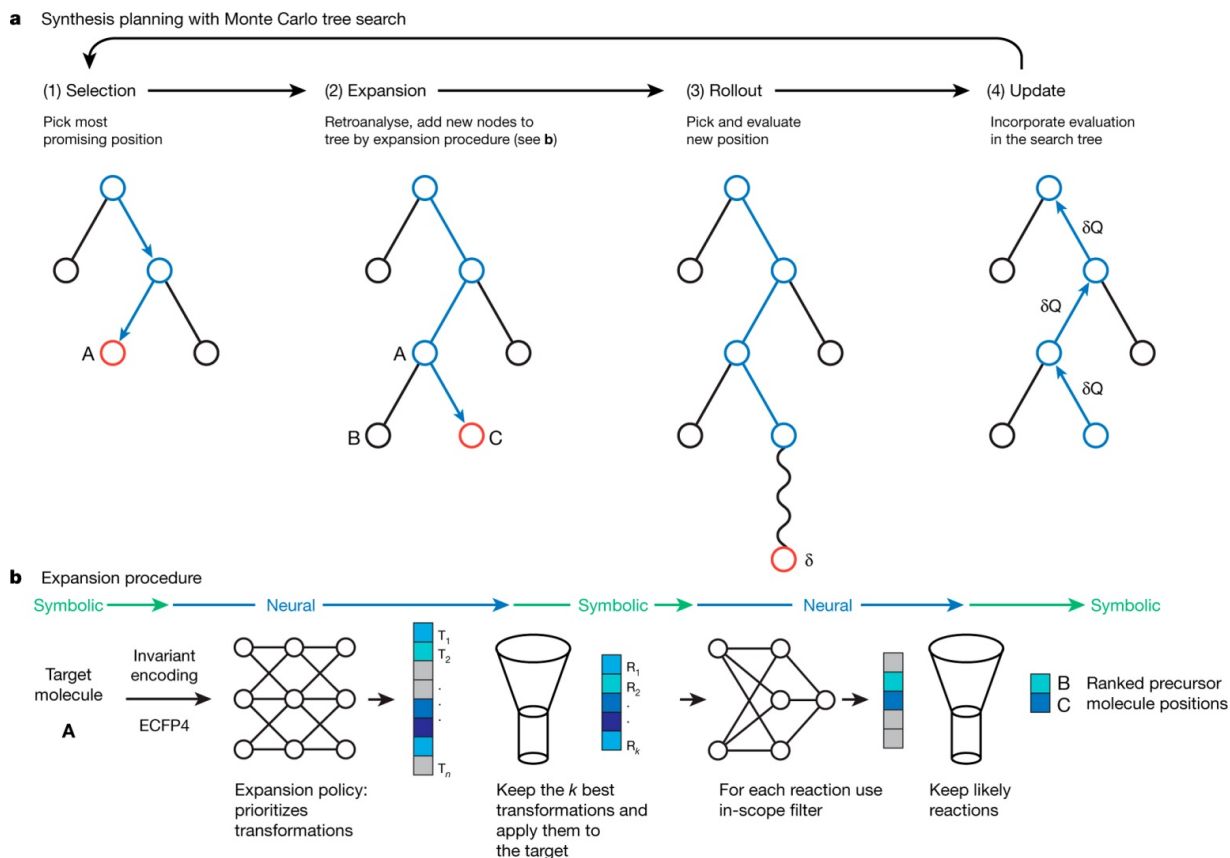
3N-MCTS on koneoppimismalli, jonka on kehittänyt Marwin et al. Se etsii retrosynteesipolkuja yksinkertaisempiin ja saatavilla oleviin lähtöaineisiin [9]. Kun retrosynteesipolku on varmennettu ja todettu toimivaksi, niin syötteenä annettu yhdiste on mahdollista syntetisoida laboratoriossa.

3N-MCTS -malli muodostuu kolmesta eri koneoppimismallista ja Monte Carlo -puuhakualgoritmista (**Monte Carlo Tree Search, MCTS**). Neuroverkot on koulutettu avustamaan puuhakualgoritmia etenemään järkevimpään suuntaan, kun haku algoritmi etsii syntetisointipolkuja ja tarkistamaan, onko ehdotettu reaktio mahdollinen kyseiselle molekyylille.

Neuroverkot ovat hakupuun laajentumisen suuntaa ohjaava verkko (**Expansion policy network, EPN**), MCTS:än rollout toimintoa tukeva Rollout -verkko (**Rollout policy network, RPN**) ja verkko, joka tarkistaa, onko syntetisointi polku toteutettavissa (**Inscope filter network, IFN**).

Data, jolla neuroverkot koulutetaan, on peräisin Reaxys -tietokannasta. Reaxys omistaa Elsevier kustantamo. Reaxys -tietokannan sisältämä data koostuu muunnossäännöistä (**transformation rule**), jotka kertovat, mitkä lähtöaineet reagoivat keskenään, mikä reaktio on kysessä ja mikä on reaktion lopputuote. Näitä sääntöjä käytetään mallien kouluttamiseen. Tietokanta sisältää yli 12.4 miljoonaa sääntöä. Mallien kouluttamiseen käytetyt säännöt sisältävät vain yksivaiheisia kemiallisia reaktioita ja reaktiossa on mukana vain yhdestä kolmeen lähtötuotetta. Eri mallien kouluttamiseen käytetään eri kriteerein suodatettua dataa tietokannasta. Kaikki koneoppimismallit koulutetaan käyttämällä ADAM -optimoijaa. ADAM -optimoija hyödyntää AdaGrad:in ja RMSProp:in toiminnallisuutta [7].

3N-MCTS toimintamalli



Kuva 4.2: 3N-MCTS:än toiminta periaate kuvattuna

[9]

3N-MCTS -mallin IFN ja EPN on kehitetty toimimaan yhdessä. Tutkittaessa puun tilaa S_i (selitä Si vaihe) jokainen molekyyli syötetään EPN:älle ja se tulostaa, mitkä säännöt voivat muodostaa annetun yhdisteen ja näin ollen myös mitkä lähtöaineet voivat muodostaa annetun yhdisteen. Nämä säännöt syötetään IFN:älle, joka suodattaa valituista reaktioista toteutettavissa olevat vaihtoehdot. Tämän jälkeen algoritmista iteroidaan neljää vaihetta, jotka muodostavat lopullisen puun.

(1) Ensimmäisessä vaiheessa algoritmi valitsee seuraavan lupaavimman tilan puusta kunnes puun lehti on saavutettu. Jos lehdessä käydään ensimmäisen kerran valintavaiheen aikana, niin lehti arvostellaan simuloimalla hakualgoritmia d askelta eteenpäin samalla muodostaen synteesipolkua (rollout). Jos lehdessä käydään useamman kuin yhden kerran valintavaiheen aikana, niin mahdolliset säännöt, jotka muodostavat lehden, tutkitaan ja lisätään lehden lapsiksi (expansion).

(2) Toisessa vaiheessa lupaavien tilojen lapset tutkitaan. Tällöin etsitään lupaavimmat

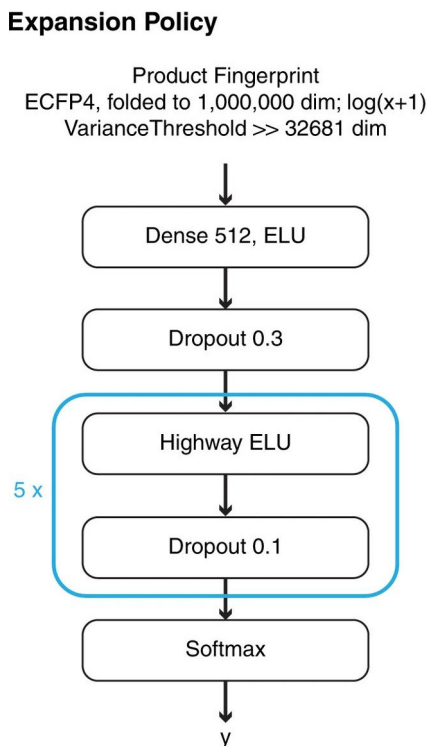
säännöt, jotka muodostavat kyseisessä tilassa olevan yhdisteen.

(3) Kolmannessa vaiheessa tarkistetaan lehden tila. Jos lehti on ‘todistettusti toimiva’, niin algoritmi palauttaa luvun suuremman kuin yksi, jolloin lehteä suositellaan käytettävän synteesipolussa. Muussa tapauksessa lehdelle suoritetaan rollout, jolloin RPN antaa rekursiivisesti uusia sääntöjä niin kauan, kunnes lehti on pilkottu lähtöaineisiin tai kunnes suurin sallittu syvyys d on saavutettu.

(4) Viimeisessä vaiheessa lehtien arvot päivitetään. Jos lähtöaineet löydetään rolloutin aikana, niin lehti saa palkinnoksi arvon 1. Jos Kaikkia lähtöaineita ei löydetty, niin lehdelle annetaan osittainen palkinto. Jos yhtään lähtöainetta ei löytynyt, niin lehti saa arvon -1.

Saattaa kuitenkin olla, että synteesipolkua ei voida luoda. Joko synteesipolun tutkimiseen menee liian kauan aikaa tai synteesipolku sisältää liian monta vaihetta yhdisteen syntetisoimiseen.

Expansion policy network



Kuva 4.3: Kuvassa expansion policy -neuroverkon rakenne.

[9]

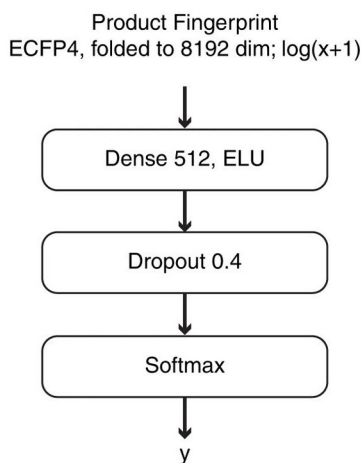
EPN muodostuu täysin kytketystä tasosta ja viidestä highway -tasosta. Molemmissa ta-

soissa on käytetty aktivointifunktiona ELU -funktiota (**Exponential Linear Unit**). Täysin kytketyssä tasossa suoritetaan dropout, jonka jälkeen vain 0.3 ensimmäisen tason solmujen tuloksista otetaan huomioon seuraavalla tasolla. Tämän jälkeen jokaisen highway-tason jälkeen suoritetaan 0.1 dropout. EPN -verkon viimeinen taso on softmax, joka palauttaa todennäköisyysjakauman. Tämä jakauma kertoo kuinka todennäköisesti muunnossääntö a muodostaa tuotteen m .

EPN -verkon kouluttamiseen valittiin datasta vain reaktiokeskus. Datasta suodatettiin pois sellaiset säännöt, jotka ilmenivät datassa alle kolme kertaa ennen vuotta 2015. Lopullinen reaktiomäärä, jolla EPN koulutettiin, oli 301 671 reaktiota.

Rollout policy network

Rollout Policy



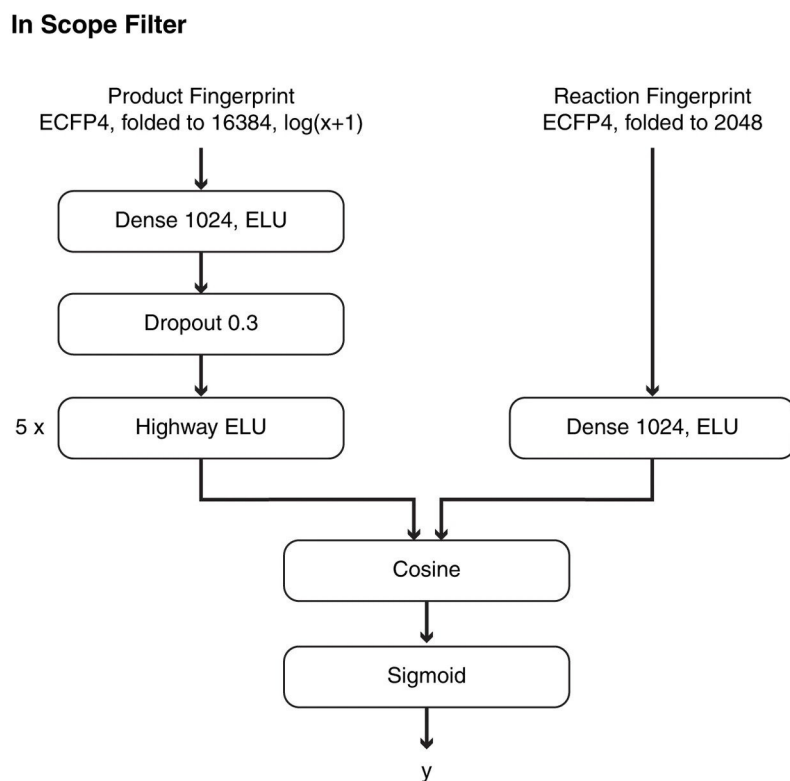
Kuva 4.4: Kuvassa Rollout policy -neuroverkon rakenne.

[9]

RPN muodostuu yhdestä täysin kytketystä tasosta. Samalla tavalla kuin EPN, RPN käyttää myös ELU -funktiota aktivointifunktiona. RPN -verkon viimeinen taso on myös softmax, joka palauttaa todennäköisyysjakauman.

RPN -verkon kouluttamiseen valittiin datasta vain reaktiossa muuttuneet atomit ja liitokset (reaktiokeskus) ja lähimmät vierekkäiset atomit. Datasta suodatettiin pois sellaiset säännöt, jotka ilmaantuivat alle 50 kertaa ennen vuotta 2015. Lopullinen reaktiomäärä, jolla RPN koulutettiin, oli 17 134 reaktiota.

In-scope filter network



Kuva 4.5: Kuvassa In scope filter -neuroverkon rakenne.

[9]

3N-MCTS:ässä IFN toimii binääriluokittelijana. IFN kertoo, onko sille annettu yhdiste mahdollista muodostaa annetulla muunnossäännöllä.

IFN -neuroverkko muodostuu kahdesta haarasta. Toinen haara saa syötteenä muodostettavan yhdisteen tunnisteiden ja toinen haara saa muunnossäännön tunnisteiden.

Haara, joka saa syötteenä yhdisteen tunnisteiden, muodostuu yhdestä täysin kytketystä tasosta jonka jälkeen suoritetaan 0.3 dropout. Tämän jälkeen data kulkee viiden highway-tason läpi. Sekä täysin kytketty että highway-tasot käyttävät ELU-funktiota aktivointifunktiona.

Haara, joka saa syötteenään muunnossäännön tunnisteiden, muodostuu vain yhdestä täysin kytketystä tasosta. Tämä taso käyttää myös ELU-aktivointifunktiota.

Näiden haarojen jälkeen saaduista tuloksista lasketaan niiden kosiniamankaltaisuus. Saatua tulos syötetään sitten sigmoid-funktiolle, jonka tulos kertoo, pystyykö kyseinen muunnossääntö muodostamaan halutun yhdisteen.

IFN koulutetaan sekä onnistuneiden että epäonnistuiden sääntöjen avulla. Koska epäonnistuneita sääntöjä ei tallenneta tietokantaan, niin kyseinen data generoidaan. Data generoidaan siten, että jos reaktiossa



lähtöaineet A ja B muodostavat reaktiossa lopputuotteen C, niin lopputuotteita D, E, F, jne.ei muodostu (voisi selittää syvemmin). IFN kouluttamista varten luotiin 100 miljoonaa epäonnistunutta reaktiota ja 10 miljoonaa epäonnistunutta reaktiota luotiin testaamista varten.

3N-MCTS suorituskyky

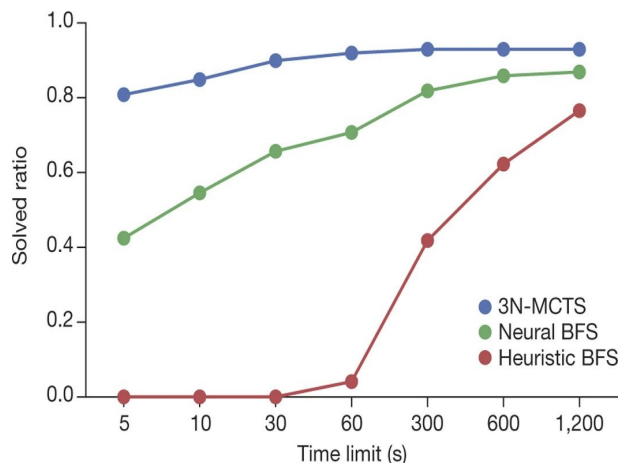
3N-MCTS -mallin suorituskykyä verrattiin BFS -menetelmään (Best-first search -method), joka käyttää heuristista käsinkoodattua $SMILES^{3/2}$ -kustannefunktioita. Tämä funktio antaa parhaimman tuloksen muunnossäännölle, joka jakaa halutun yhdisteen mahdollisimman hyvin yhtä suuriin lähtötuotteisiin. Tästä menetelmästä luotiin myös versio, joka käyttää muokattua $SMILES^{3/2}$ -kustannefunktioita. Muokattu funktio käyttää 3N-MCTS:än expansion policy -neuroverkkoa ja in-scope filter -neuroverkkoa parhaimman muunnossäännön selvittämiseksi.

Mallit koulutettiin samoilla muunnossäännöillä ja testeissä käytettiin samaa aikaan perustuvaa jakoa. Mallit koulutettiin muunnossäännöillä, jotka ilmenivät ennen vuotta 2015, ja malleja testattiin vuoden 2015 jälkeen ilmenneillä muunnossäännöillä.

Mallien tehtäväksi annettiin löytää halutulle yhdisteelle retrosyntetisointipolku annetussa ajassa. Testaamiseen käytettiin 497 eri yhdistettä.

3N-MCTS suoriutui kahta muuta algoritmia paremmin jokaisessa eri aikakategoriassa. 3N-MCTS löysi viidessä sekunnissa 80 prosentille yhdisteistä retrosyntetisointipolun, kun taas muokattu $SMILES^{3/2}$ -menetelmä löysi retrosyntetisointipolun vain 40 prosenttia yhdisteistä ja alkuperäinen ei löytänyt polkuja ollenkaan. Kun aikaa kasvatettiin 60 sekuntiin per yhdiste, 3N-MCTS löysi 92 prosentille yhdisteistä polun, muokattu $SMILES^{3/2}$ löysi 71 prosentille yhdisteistä ja alkuperäinen versio vain 4 prosentille yhdisteistä.

3N-MCTS:ää testattiin myös sokkotestillä 45 yliopistotason kemistille. Testin tarkoituksena oli selvittää, suosivatko henkilöt enemmän kirjallisuuteen perustuvaa retrosyntetisointipolkua vai 3N-MCTS:än antamaa polkua. Sokkotestissä 57 prosenttia henkilöistä valitsi 3N-MCTS muodostaman polun ja loput valitsivat kirjallisuuteen perustuvan polun.



Kuva 4.6: 3N-MCTS:än suorituskky verrattuna $SMILES^{3/2}$ hyödyntävään BFS -menetelmään ja 3N-MCTS:än neuroverkkoja hyödyntävään BFS -menetelmään. Kuvaaja kertoo, kuinka monelle prosenttia annetuista yhdisteistä menetelmä kykenee löytämään retrosyntetisointipolun annatussa ajassa. Annetulla ajalla tarkoitetaan aikaa, jonka menetelmä voi korkeintaan käyttää yhden yhdisteen retrosyntetisointipolun löytämiseen.

[9]

4.3.2 Heurustiikka avusteiset neuroverkot

On myös yleistä, että retrosynteesipolkuja suunnitelevat mallit hyödyntävät heurustiikka [1]. Badowski et al. ryhmä on kehittänyt koneoppimismallin, joka suunnittelee retrosynteesipolkuja hyödyntämällä neuroverkkoja ja ammattikemistien muodostamia heuristisia sääntöjä suunnitellessaan retrosynteesipolkua.

ICHO:n (Instytut Chemii Organicznej) kouluttamiseen käytettiin dataa 1.4 miljoonasta reaktiosta ja niiden lopputuotteista. Data saadaan julkaistusta artikkeleista ja patenteista. Reaktiolta kuitenkin vaaditaan, että ainakin yksi reaktio per lopputuote löytyy myös Chematican (taustatietoa) expert-coded säännöstöstä. Lisäksi datasta suodatettiin pois suojaryhmien säännöt.

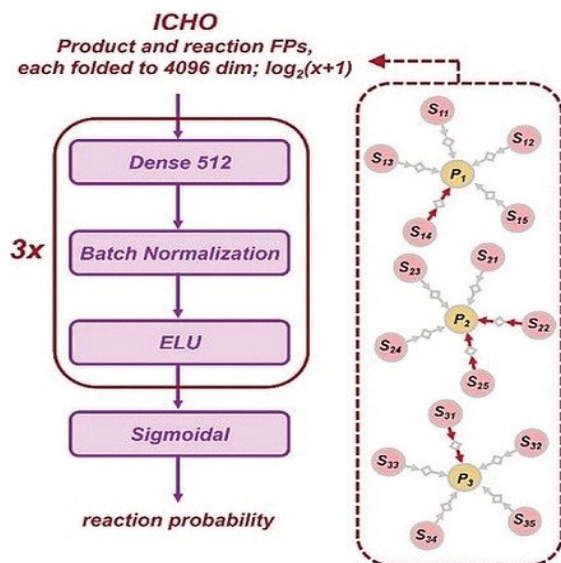
ICHO:sta kehitettiin myös malli, jonka kouluttamisessa käytettiin lisäksi vektoreita, jotka sisälsivät dataa kemiallisesti intuitiivisista reaktiopiirteistä, joita on käytetty aikaisemmissa tutkimuksissa. Tätä mallia kutsutaan ICHO+ malliksi, ja sitä käytetään vertailemaan ICHO mallin suoriutumista.

Kehitettyssä ICHO mallissa on kaksi tärkeää asiaa.

(1) Se laskee reaktioiden ilmaantumistodennäköisyyttä uudelleen perustaen kuinka usein ne ilmaantuvat ammattilaisten käyttämissä reaktioissa. Tämä laskenta tapahtuu mallin

koulutusvaiheessa. Käytännössä tämä tarkoittaa, että koulutusvaiheessa mallilla on tieto, kuinka monta kertaa tietty reaktiosääntö ilmeni datassa ja kuinka monta kertaa kyseistä reaktiota käytettiin luomaan jokin lopputuote. Malli laskee näiden välisen suhteen, jolloin se pystyy määrittämään, kuinka usein kyseistä reaktiota kannattaa käyttää. Jos esimerkiksi jokin reaktio on mainittu datassa kymmenen kertaa ja samassa datassa kyseistä reaktiota käytetään kymmenen kertaa luomaan jokin lopputuote, niin malli käyttää tätä reaktiota, koska se luokitellaan ‘Helpoksi ja varmaksi toteuttaa’. Jos suhde taas on pieni, niin reaktio luokitellaan ‘vaikeaksi toteuttaa’.

(2) Toiseksi se kykenee antamaan suuremman kuin nollatodennäköisyyden reaktiolle, joka ei ilmaantunut koulutusvaiheessa. Tämä sen takia, että mallille koulutetaan jatkuva funktio, joka antaa todennäköisyyden jokaiselle reaktiotyypille perustuen, kuinka reaktio muuttaa lopputuotteen lähtötuotteiksi. Eli jos jokin reaktion sormenjälki (Lisää tietoa, koska useassa mallissa käytetään) tai heuristinen kuvaus on samankaltainen toisen reaktion kanssa, joka ilmenee koulutusdatassa, niin reaktiolle voidaan antaa todennäköisyys sen käytölle.



Kuva 4.7: ICHO -neuroverkon rakenne.

[1]

ICHO -malli muodostuu neljästä kerroksesta, joista kolme kerrosta ovat täysin kytkettyjä neuroverkkoja ja viimeinen taso on sigmoid -taso. Samalla tavalla kuin 3N-MCTS mallissa, ICHO -malli käyttää myös ELU -aktivointifunktiota. ICHO -malli käyttää Batch normalization -algoritmia koulutusvaiheessa. Batch normalization -algoritmin idea on standardoida

koulutusvaiheessa kerrokselle tuleva syöte, jolla verkon painoja muutetaan.

ICHO mallia vertailtiin aikasemmin mainitun 3N-MCTS -mallin kanssa, jota kutsutaan artikkelissa SW mallina. Kyseisestä SW mallista kehitettiin myös heuristista dataa hyödyntävä malli SW+. Lisäksi luotiin SW malli, joka ei käy läpi kaikkia mahdollisia reaktioita, vaan valitsee säännöt niistä reaktioista, jotka eivät aiheuta konfliktia (selitys) ja jotka johtavat vain tiettyyn lopputuotteeseen. Tätä mallia kutsutaan SW2 -malliksi ja tästä luodusta heuristisesta mallista käytetään nimeä SW2+. Lisäksi vertailumalliksi luodaan täysin heuristinen malli, joka arvioi reaktioita sen mukaan, kuinka paljon ne rakenteellisesti yksinkertaistavat lopputuotetta. Tämä malli suosii reaktioita, jotka pilkkovat molekyylin keskeltä, eli puolittavat sen. Tämä on haluttu lähestymistapa retrosynteesiin, koska se minimoi syntetisointivaiheiden määrän.

ICHO suorituskky

ICHO -mallia testataan eri tavalla kuin 3N-MCTS mallia, jossa mitataan, kuinka monelle yhdisteelle malli kykenee löytämään toimivan retrosyntetisointipolun tietyssä ajassa. IC-HO -mallia testataan siten, että jokaiselle mallille annetaan yhdiste, ja malli antaa tulosteena listan muunnossäännöistä, jotka on asetettu paremmuusjärjestykseen niille annetun pisteytyksen mukaan. Muunnossäänön pisteytys kertoo, kuinka paljon malli suosittelee käytettävän kyseistä muunnossääntöä. Näistä muunnossäännöistä laskettiin, kuinka suuri osa niistä pohjautuu johonkin aikaisempaan muunnossääntöön, joka on esiintynyt jossain kirjallisuudessa. Mitä suurempi osa ehdotetuista muunnossäännöistä löytyy kirjallisuudesta, sitä paremmin malli toimii.

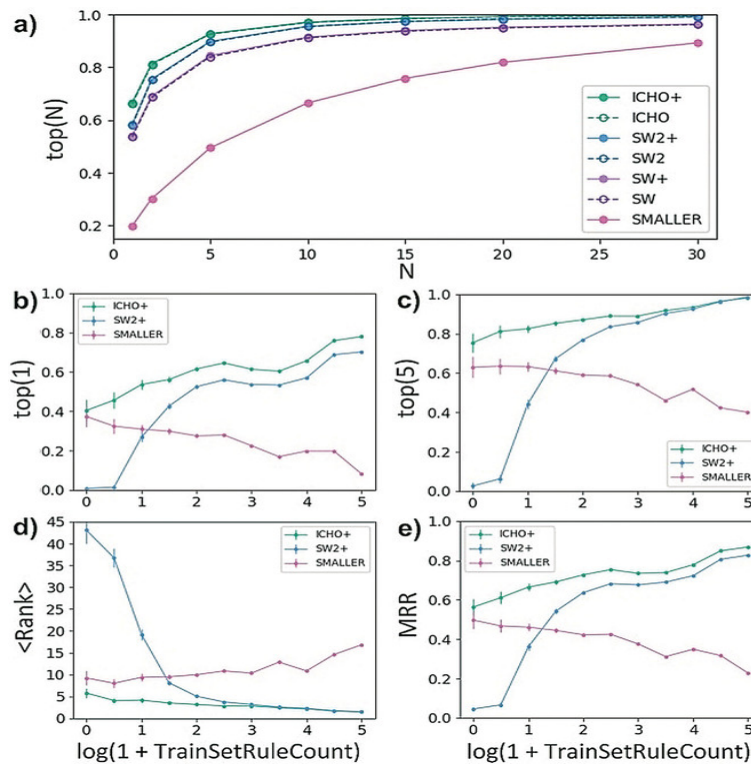
Kuvaajaan on piirretty vain eri mallien heuristiikkaa hyödyntävät versiot, koska ne pärjäsivät testeissä ei-heuristisia versioita paremmin. Lisäksi kuvaajaan on piirretty *SMALLER* malli vertailuksi.

Nähdään, että ICCHO+ -malli pisteyttää kirjallisuuteen perustuvat muunnossäännöt paremmin kuin muut mallit, vaikkakin eroavaisuudet ovat vähäiset.

Kuvaajista b ja c nähdään, että pienelläkin koulutusaineistolla ICCHO+ sijoittaa kirjallisuuden muunnossäännöt korkeammalle kuin muut mallit ja mitä enemmän koulutusaineistoa käytetään mallin kouluttamiseen sitä suurempi osa korkeimmin sijoitetuista muunnossäännöistä perustuvat kirjallisuuteen. Mielenkiintoinen huomio, mikä kuvaajasta nähdään, että *SMALLER* -malli sijoittaa kirjallisuuteen perustuvat säännöt pienemmällä koulutusaineistolla korkeammalle kuin suuremmalla aineistomäärällä. SW2+ -malli taas toi-

mii päinvastaisesti. Mitä enemmän koulutusaineistoa se saa sitä korkeammalla se sijoittaa kirjallisuuteen perustuvt säännöt.

Kuvaajat d ja e eivät ole ristiriidassa kuvaajien a-c kanssa. Kuvaajast d nähdään, että nopeasti koulutusaineiston kasvaessa tutkittavat muunnossäännöt siirtyvät nopeasti korkeille sijoille sw2+ -mallissa ja ICHO+ -malli sijoittaa ne korkealla jo pienellä data määrällä. *SMALLER* -malli sen sijaan sijoittaa tutkittavat muunnossäännöt koulutusaineiston kasvaessa huonommalle sijalle. Kuvaaja e myös näyttää, että ICHO+ j SW2+ -mallien MRR -pisteet kasvavat koulutusaineiston kasvaessa, kun taas *SMALLER* -mallin pisteytys laskee.



Kuva 4.8: ICHO -neuroverkon ja muiden testattujen koneoppimismallien suorituskyyky kuvattuna. Kuvaajat a-c kertovat, kuinka suuri osa n-määrästä parhaiten pisteytettyjä muunnossääntöjä oli jostain kirjallisuudesta, esim. raportti. Esim, top 1 = 1 tarkoittaa, että malli antaa parhaimman pisteytyksen aina kirjallisuuteen perustuvalla muunnossäännöllä. Kuvaaja d kertoo, mille sijalle kukin malli sijoittaa tietyn muunnossäännön. Tässä tapauksessa rank = 1 tarkoittaa, että malli sijoittaa tietyn muunnossäännön aina sijalle yksi. Kuvaaja e kertoo keskiarvon kirjallisuuteen perustuvien muunnossääntöjen käänteisestä sijoituksesta. Tämä tarkoittaa, että jos malli sijoittaa kaikkien kirjallisuuteen perustuvat muunnossäännöt ensimmäiseksi, niin MRR saa arvon yksi.

[1]

Miten koneoppimista hyödynnetään tällä hetkellä lääkkeiden syntetisoinnissa [9, 6, 10]?

5 Lääkkeiden kehitys tulevaisuudessa

Vaikka koneoppimisen käyttökohteet ovat yleistyneet lääkekehityksessä, niin koneoppimisen lähestymistavoilla on edelleen potentiaalia kehittyä [3].

Yksi kehityksen kohteista on tehokkaampi datan tulkinta pienestä aineistomäärästä. Paikoin kemian ja lääkekehityksen alalla on muihin koneoppimismallien käyttökohteisiin verrattaen vähän dataa saatavilla ja sitä on kallista tuottaa. Tämä vaatii tutkijoilta enemmän työtä, jotta tieteellisistä julkaisuista saatava data saadaan koneiden hyödyntämään muotoon. Toinen ratkaisu tähän on meta-oppimis (selitys) lähestymistapa. Uudet lähestymistavat kuten Neuro Turing kone ja matkija oppiminen mahdollistavatkin oppimisen vähästä määrästä dataa ja Bayes luokittelija on suoriutunut lähes ihmistasoisesti käyttämällä One-shot oppismallia.

Toinen merkittävä edistysaskel on kemiallisten yhdisteiden ja reaktioiden tehokkaampi esitysmalli. Tähän asti kemiaa on esitetty ihmisten ymmärtämässä muodossa, mutta tämä ei aina ole koneelle paras esitysmalli. Koneoppimismallit käyttävät tietoa molekyyleistä ja atomeista ja koneoppimismallit ovat niin hyviä kuin nämä kuvaukset. Hyvältä kuvaukselta vaaditaan, että sen avulla on yksinkertaista hankkia kohteen ominaisuudet ja sen tulee olla mahdollisimman pieni ulotteinen. Uusia kuvausmalleja on kehitetty ja koska uudet mallit ovat näyttäneet, että ne ovat tehokkaita, niin uusien kuvauksien kehittäminen jatkuu edelleen.

Kemiallista koneoppimista edistävä asia on myös kvanttilaskennan hyödyntäminen. Kvanttikoneiden suuri laskuteho pystyy vähentämään uusien mallien koulutusaikaa merkittävästi.

Koneoppimista voidaan mahdollisesti käyttää tulevaisuudessa uusien tieteellisten lakien löytämisessä. Mutta vaikka eri koneoppimisesta kehitetyt mallit ovat ennalta arvattavia, niin ne eivät kuitenkaan ole aina tulkittavia. Neuroverkko voi esimerkiksi oppia ideaalin kaasulain ($pV = nRT$), mutta neuroverkon kaarien painojen muuttaminen ymmärrettäväksi säännöksi ei ole yksinkertainen tehtävä. Voi olla, että koneoppimismalli pystyy huomaamaan datasta säännön, joka toistuu, mutta jos tutkijat eivät tiedä tai tunne kyseistä sääntöä, niin mallin tulkitseminen on lähes mahdotonta.

6 Yhteenveto

Lähteet

- [1] T. Badowski, E. P. Gajewska, K. Molga ja B. A. Grzybowski. "Synergy Between Expert and Machine-Learning Approaches Allows for Improved Retrosynthetic Planning". *Angewandte Chemie International Edition* 59.2 (2020), s. 725–730. DOI: <https://doi.org/10.1002/anie.201912083>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/anie.201912083>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.201912083>.
- [2] G. R. Bickerton, G. V. Paolini, J. Besnard, S. Muresan ja A. L. Hopkins. "Quantifying the chemical beauty of drugs". eng. *Nature chemistry* 4.2 (2012), s. 90–98. ISSN: 1755-4330.
- [3] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev ja A. Walsh. "Machine learning for molecular and materials science". eng. *Nature (London)* 559.7715 (2018), s. 547–555. ISSN: 0028-0836.
- [4] C. W. Coley, W. H. Green ja K. F. Jensen. "Machine Learning in Computer-Aided Synthesis Planning". eng. *Accounts of chemical research* 51.5 (2018), s. 1281–1289. ISSN: 0001-4842.
- [5] S. Ekins, A. C. Puhl, K. M. Zorn, T. R. Lane, D. P. Russo, J. J. Klein, A. J. Hickey ja A. M. Clark. "Exploiting machine learning for end-to-end drug discovery and development". eng. *Nature materials* 18.5 (2019), s. 435–441. ISSN: 1476-1122.
- [6] S. Harel ja K. Radinsky. "Accelerating Prototype-Based Drug Discovery Using Conditional Diversity Networks". Teoksessa: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '18. London, United Kingdom: Association for Computing Machinery, 2018, s. 331–339. ISBN: 9781450355520. DOI: 10.1145/3219819.3219882. URL: <https://doi-org.libproxy.helsinki.fi/10.1145/3219819.3219882>.
- [7] D. P. Kingma ja J. Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].
- [8] M. Popova, O. Isayev ja A. Tropsha. "Deep reinforcement learning for de novo drug design". eng. *Science advances* 4.7 (2018), eaap7885–eaap7885. ISSN: 2375-2548.

- [9] M. H. S. Segler, M. Preuss ja M. P. Waller. "Planning chemical syntheses with deep neural networks and symbolic AI". eng. *Nature (London)* 555.7698 (2018), s. 604–610. ISSN: 0028-0836.
- [10] B. Shin, S. Park, J. Bak ja J. C. Ho. "Controlled Molecule Generator for Optimizing Multiple Chemical Properties". Teoksessa: *Proceedings of the Conference on Health, Inference, and Learning*. CHIL '21. Virtual Event, USA: Association for Computing Machinery, 2021, s. 146–153. ISBN: 9781450383592. DOI: 10.1145/3450439.3451879. URL: <https://doi.org/10.1145/3450439.3451879>.
- [11] C. Sottriffer, R. Mannhold, H. Kubinyi ja G. Folkers. *Virtual Screening: Principles, Challenges, and Practical Guidelines*. eng. Vol. 48. Methods and principles in medicinal chemistry. Weinheim: John Wiley ja Sons, Incorporated, 2011. ISBN: 9783527326365.
- [12] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer ja S. Zhao. "Applications of machine learning in drug discovery and development". eng. *Nature reviews. Drug discovery* 18.6 (2019), s. 463–477. ISSN: 1474-1776.
- [13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser ja I. Polosukhin. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].
- [14] D. Weininger. "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules". *Journal of Chemical Information and Computer Sciences* 28.1 (1988), s. 31–36. DOI: 10.1021/ci00057a005. eprint: <https://pubs.acs.org/doi/pdf/10.1021/ci00057a005>. URL: <https://pubs.acs.org/doi/abs/10.1021/ci00057a005>.