TARTU ÜLIKOOL
HUMANITAARTEADUSTE JA KUNSTIDE VALDKOND
EESTI JA ÜLDKEELETEADUSE INSTITUUT

Linda Freienthal

PRONOMINAALSETE VIITESUHETE AUTOMAATNE LAHENDAMINE EESTI KEELES NÄRVIVÕRKUDE ABIL

Magistritöö

Juhendaja dotsent Kadri Muischnek

Tartu 2020

SISUKORD

SISSEJUHATUS

Arvutilingvistid ja keeletehnoloogid tegelevad igapäevaselt vabateksti automaatse mõistmisega. Selleks, et teksti automaatselt mõista ja töödelda, on vaja võimalikult palju informatsiooni antud dokumendi kohta. Vaid sõnade süntaktilisest ja morfoloogilisest infost ei piisa näiteks juhul, kui tõlkida automaatselt eesti keelest vene või inglise keelde sõna *tema*, kuna viimastel on kohustuslik määrata ka *tema* sugu (*he/she* või *он/она*). Samuti hõlbustaks efektiivset infoeraldust teave, millised sõnad viitavalt tegelikult ühele ja samale olemile. Kui me saaksime tekstist kätte iga pronoomeni tegelikult tähendused, paraneks oluliselt näiteks automaatsete sisukokkuvõtjate kvaliteet: kontekstist välja rebitud tähtsamates lausetes saab asendada segadust tekitavad pronoomenid nende tegelike tähendustega. (Mitkov 2002: 275–276)

Kuna taoline samaviiteliste sõnade automaatne leidmine on oluline alus paljudele keeletehnoloogilistel vahenditele, on **viitesuhete automaatset lahendamist** (ingl *coreference resolution*) peetud 1960ndatest saadik (Ng 2017: 1, Stuckardt 2016: 1) loomuliku keele töötluse üheks põhiprobleemiks. **Viitesuhe** koosneb kahest osast: **asendussõnast** ja **viitealusest** (ingl *antecedent*), millele ta viitab. Alltoodud näites on *ta* pronominaalne asendussõna, mis viitab oma viitealusele *Linda*. Nende vahel on viitesuhe.

Linda on pärit Lääne-Virumaalt. Nüüd elab *ta* Tartus.

Viitesuhetest saab moodustada ka **viitesuhete keti** (ingl *coreference chain*). Kui lisada ülaltoodud näitele lause *Ning* tal *on kilpkonn ja kaks kassi*, siis moodustaksid antud tekstis viiteahela sõnad *Linda, ta* ja *tal*. Need kõik viitavad ühele ja samale olemile (reaalsele objektile) ja on omavahel viitesuhtes. Üldjuhul peetakse viitealuseks asendussõnast eespool asuvat sõnet. Sel juhul on tegemist anafooride[1] lahendamisega (ingl *anaphora resolution*).

Kui antud viiteahela või -suhte info automaatselt kätte saada, saaks masintõlkijale anda info, et *ta* ja *tal* on naissoost, sisukokkuvõtja jaoks asendada pronoomenid pärisnimega ning anda infoeraldajale teada, millised laused käivad tegelikult *Linda* kohta.

[1] Anafoor on tagasiviide.

Pronoomenite automaatne lahendamine aitab näiteks ka küsimustele vastamise süsteemide kvaliteeti tõsta (Vicedo, Ferrández 2000).

Eesti keeles on töö autorile teadaolevalt tegelenud viitesuhete automaatse lahendamisega neli inimest. Pilleriin Mutso (2008) kohandas oma magistritöös Mitkovi teadmistevaest[2] reeglipõhist asendussõnade lahendajat (Mitkov 2002) eesti keele pronoomenitele *tema* ja *nemad*, otsides vaid anafoorseid viitesuhteid (st, ta otsib võimalikke viitealuseid vaid pronoomenile eelnevate sõnade seast). Mutso lahendaja suutis lahendada alla 74% viitesuhetest. Ka Tiina Puolakainen (2015) lähenes oma katsetustes teadmistevaeselt ja reeglipõhiselt, kuid tema kasutas kitsenduste grammatikat (ingl *constraint grammar*) ja otsis nii anafoorseid kui ka katafoorseid pronominaalseid viitesuhteid. Puolakainen suutis lahendada 70–79% viitesuhetest. Ei Mutso ega Puolakaineni töö tulemused pole teiste arvutilingvistide ja keeletehnoloogide kasutusse jõudnud ja on jäänud vaid katsetusteks.

Aastatel 2015–2017 loodi projekti „Sihipärane süntaks korpuse jaoks“ raames ca 107000 tekstisõna suurune ajalehetekstide korpus, kus on pronominaalsetele asendussõnadele *mina-meie, sina-teie, tema-nemad, kes, mis* ja *see-need* viitealus(t)e olemasolu korral need käsitsi märgendatud. Freienthal (2018) uuris oma bakalaureusetöös selles korpuses olevaid pronominaalseid viitesuhteid, et leida seaduspärasusi, mida saaks pronominaalsete viitesuhete lahendaja loomisel ära kasutada. Bakalaureusetöös leitud reeglites ja seaduspärasustes oli näha, et need katsid üle 75% viitesuhetest, jättes automaatselt arvestamisväärt koguse viitesuhetest välja, ja lootus saavutada reeglipõhise lahendajaga märkimisväärseid tulemusi on väike. Seetõttu läheneb antud töö bakalaureusetöö jätkuna pronominaalsete viitesuhete lahendamisele närvivõrkude meetodil.

2020. aasta maikuu seisuga ei ole veel ilmunud Tartu Ülikooli keeletehnoloogia teaduri Eduard Barbu, keeletehnoloogia vanemteaduri ja arvutilingvistika dotsendi Kadri Muischneki ning töö autori koostööna valminud ülalmainitud korpusel[3] treenitud

[2] Teadmistevaene tähendab vähese morfoloogilise ja süntaktilise infoga, jättes semantika kõrvale.
[3] Mida hiljem suurendati ca 147000 sõne võrra.

masinõppepõhiste automaatsetete lahendajate tulemused. Eduard Barbu keskendus mitte-närvivõrk meetoditele. Selle magistritöö eesmärk on lahendada närvivõrkudega pronominaalseid viitesuhteid, kasutades osutus-paari mudelit (vt lk 10). Mõlema töö aluseks olnud korpuse ja treenimismaterjali loome kohta saab täpsemalt lugeda peatükkides 3.1 ja 3.2.

Töö koosneb kolmest osast. Esimeses osas tutvustatakse tähtsamaid mõisteid antud töös ja antakse ülevaade asendussõnade automaatsete lahendajate mudelitüüpidest, mida on seni teistes keeltes katsetatud, keskendudes mõnele olulisemale tööle. Teises osas tutvustatakse korpust ja selle tähtsamaid aspekte lahendaja loomise kontekstis. Samuti kirjeldatakse korpuse eeltöötlust ja treeningmaterjali, millel põhinevad kõik katsetatud mudelid. Kolmandas osas kirjeldatakse parimaid närvivõrke ja analüüsitakse nende tulemusi.

1. ASENDUSSÕNADE AUTOMAATSE LAHENDAMISE VIISID MASINÕPPE MEETODITEL

1.1. Tähtsamad mõisted

Viitesuhete tüüpe keeleteaduslikust vaatepunktist lahkab autor oma bakalaureusetöös (Freienthal 2018: 6–8). Arvutilingvistide sõnavaras on olulised lisaks sissejuhatuses kirjeldatud mõistetele *asendussõna* (mida üldiselt piiratakse pronoomenitega), *viitealus* ja *viitesuhe* ka mõisted *osutus* (ingl *mention*), *olem* (ingl *entity*).

Olem on reaalses füüsilises maailmas eksisteeriv objekt või ka abstraktne mõiste. **Osutus** on fraas või sõna, millega viidatakse olemile. Ühes tekstis võib olemile viidata mitu erinevat osutust. Näites (1) viitab olemile, päris inimesele, kes on Eesti Vabariigi president, kolm erinevat osutust – *President Kersti Kaljulaid, Ta, Kerstile*. Osutuste eraldamist viitealuste kandidaatidena (mida hiljem masinõppe mudelile sisse sööta) teeb keerulisemaks asjaolu, et mõni fraas võib sisaldada mitu osutust. Näidetes (2) ja (3) on viiesõnaline fraas, mis viitab naabrinaise koera kuudile. Näites (2) viitab asendussõna *See* kogu fraasile, kuid näites (3) viitab asendussõna *tal* hoopis selle omajale, sõbralikule koerale. Taoline fraaside automaatne tükeldamine võimalikeks osutusteks, mida pidada viitealuste kandidaatideks, on keeruline ülesanne.

(1) *President Kersti Kaljulaid* jõudis Antarktikasse. *Ta* osaleb Antarktika avastamise 200. aastapäeva tähistamisel. *Kerstile* meeldib matkata.
(2) *Naabrinaise sõbraliku koera kollasel kuudil* laseb katus läbi. *See* läheb lammutamisele.
(3) *Naabrinaise sõbraliku koera* kollasel kuudil laseb katus läbi. Nüüd *tal* polegi kuiva kohta, kus magada.

Selle teooriaosa mõistmist hõlbustab üldine arusaam masinõppe tööst. **Masinõpe** on teadusala, mis tegeleb algoritmidega ja statistiliste mudelitega, mis õpivad ise andmestiku pealt väljundit looma. See tähendab, leiab ise otsustusreeglid. **Mudel** on algoritm, mis on juba treenitud andmestiku peal. See tähendab, on näinud andmestikku ja selle peal oma reegleid õppinud ja oskab uut andmestiku nähes ennustada, milline peaks väljund olema. Selleks, et mudelit treenida, on vaja talle sisendit. Sisendiks on

treenimisüksused ehk näited. Neid näiteid esitatakse tunnuste vektoritena. Juhendatud masinõppes lisatakse vektorisse ka soovitud väljund, mida mudel lõpuks ennustama peab hakkama. **Vektorist** võib mõelda kui arvujadast, kuigi tegelikult on see mitmedimensiooniline esitus treenimisüksusest. Arvud vektori sees on algoritmile sobivale kujule tõlgitud **tunnused**, mille põhjal algoritm oma reeglid ehk mudeli loob. Hiljem saab mudelile sisse sööta uusi, seni nägemata üksuste vektoreid ja näha, millise väljundi mudel välja arvutab.

1.2. Lühike ülevaade asendussõnade lahendajate arengust

Asendussõnade automaatsele lahendamisele on lähenetud alati vastavalt võimalustele ja ressurssidele, mis antud hetkel saadaval on. Esimesed tööd olid pigem teoreetilised, kindlale domeenile ehk tekstiliigile või kasutusvaldkonnale keskendunud, rangete reeglite põhised ja ei jõudnud tavakasutusse (Mitkov 2002: 68). 1980ndatel saadi aru, et lõpp-tarbijani jõudnud lahendused ei või olla liiga domeenikesksed ja peavad olema robustsemad, et rohkemate tekstidega hakkama saada. Reeglite otsingul hakati lähtuma pigem keeleteadusest kui domeenist. See aitas vähendada domeenispetsiifilisust. Ka reeglipõhist lähenemist arendati edasi. 1990ndateks tähendas reeglipõhine lähenemine pigem sõelasüsteemi kui sellele eelnenud karmi otsingureeglistikku. Sõelasüsteem tähendab, et algoritm koosneb välistavatest ja eelistavatest reeglitest, mis kas viskavad viitealuse kandidaadi kandidaatide hulgast välja või annavad sellele punkte. Võitja saab viitealuseks. (Stuckardt 2016: 3–5, Mitkov 2002: 68–92)

Lahendajate kasutust piiras (ja piirab siianigi) teiste parserite[4] olemasolu ja kvaliteet. See tähendab, et kui lahendaja toetub oma sõelumisel sõnade süntaktilisele ja morfoloogilisele infole, siis saab see häid tulemusi käsitsi märgendatud tekstidel, kus ka süntaks ja morfoloogia on käsitsi sõnadele juurde lisatud, kuid mitte tavatekstidel, mida süntaksi ja morfoloogia parserid ei oska veel kvaliteetselt analüüsida. 1990ndate keskel hakati sellele rohkem tähelepanu pöörama ja prooviti luua teisi parsereid mitte vajavaid lahendajaid. (Stuckardt 2016: 5, Mitkov 2002: 105–110) Praeguseks on vajalike

[4] Parser on siin tekstitöötlusvahend, mis lisab tekstile mingisugust infot. Näiteks sõnaliigi infot iga sõna kohta. Ka viitesuhete automaatset lahendajat võib parseriks nimetada.

parserite kvaliteet juba piisavalt heal tasemel, et neid kasutada (sh ka eesti keeles).

1995. aastal korraldati esimene viitesuhete lahendajate võistlus (Sundheim 1995), mille hindamiseks loodi asendussõnade suhtes käsitsi märgendatud korpus. See korpus oli esimene suurem asendussõnade suhtes märgendatud korpus inglise keeles, mille abil täiustati olemasolevaid reeglipõhiseid lähenemisi. Aga tähtsaim veelgi – see andis aluse statistika- ja masinõppepõhistele lähenemistele. (Stuckardt 2016: 6–7) Ka sellele järgnenud võistlused on oluliselt arendanud viitesuhete lahendajate tööstust.

Kahetuhandetatest tänaseni on loodud uusi ja suurendatud vanu korpuseid, arendatud korpusepõhist lähenemist, tegeletud masinõppega ning ära pole unustatud ka reeglipõhist lähenemist, luues hübriidlahendajaid, mis kasutavad nii reegleid kui ka masinõpet. Suurem fookus on reaalselt kasutatava lahendaja loomisel, mida saaks ühendada mõne teise keeletehnoloogilise vahendiga. Mõni lahendaja kasutab veel ära domeenispetsiifilisi ressursse ning arvutilingvistika varalaekasse on lisandunud sellised andmebaasid nagu Wikidata, WordNet ja OntoNotes, mille abil saavad keeletehnoloogilised vahendid ära kasutada semantilist teadmust. Samuti ei ole kõik kogu probleemi korraga ette võtnud, vaid on lähenenud ülesandele jupp-haaval, alustades näiteks selliste mudelite loomisest, mis määravad, kas antud asendussõnal on suur tõenäosus, et tal on viitealus või mitte. Sellest järgmine jupp oleks alles viitealuse määramine neile asendussõnadele, millel eelmise mudeli kohaselt suure tõenäosusega on viitealus. (Stuckardt 2016: 7–9, Lee, H. jt 2017: 6–7)

Eesti keeles on seni katsetatud reeglipõhiseid lähenemisi (Mutso 2008, Puolakainen 2015), loodud pronominaalseid viitesuhteid sisaldav ajalehetekstide korpus[5] ning seda analüüsitud reeglipõhise lähenemise kontekstis (Freienthal 2018). Selle korpuse põhjal on tehtud esimesi katseid lahendada pronominaalseid viitesuhteid masinõppe meetoditega (vt peatükk 3.9). Antud magistritöö proovib pronominaalseid viitesuhteid lahendada masinõppe alaliigi, närvivõrkude abil.

Järgnev alapeatükk tutvustab erinevaid masinõpet kasutavaid lähenemisviise lahendajatele, mida mujal maailmas on katsetatud.

[5] Korpus on kättesaadav aadressil https://github.com/EstSyntax/EstAnaphora.

1.3. Asendussõnade automaatse lahendaja mudelitüübid

Viitesuhete automaatne lahendamine on keeruline ja huvitav ülesanne, millele saab läheneda mitmel moel. Algoritme, mida kasutada, on masinõppe arsenalis palju. Varieeruvust lisavad ka erinevalt märgendatud korpused ja viisid, mil moel viitesuhet vaadelda ja masinõppe algoritmile esitada.

Üks esimesi masinõppe katsetusi ja tänini levinumaid mudelitüüpe on **osutus-paari mudel** (ingl *mention-pair model*). See esitab asendussõna lahendamise probleemi tavalise klassifitseerimisprobleemina, andes mudelile ette osutuste paari koos nendevahelise suhte morfoloogiliste, süntaktiliste jms tunnustega. Mudel tagastab binaarse vastuse, kas see paar on viitesuhtes või ei ole. Iga masinõppe algoritm saab taolise klassifitseerimisega hakkama. (Ng 2017: 2–3, Novák 2018: 24–25) Kui näide (4) oleks treeningandmestik, saaks algoritm endale ette tunnuste vektorid, millest igaüks sisaldaks ühte osutuste paari (näiteks *Mari-ta, Mari-Sipsikut, Mari-Ma* jne), koos nende osutuste ja osutustevaheliste tunnustega (näiteks morfoloogiline ja süntaktiline info, sõnadevaheline kaugus) ning väljundiga, kas nende vahel on viitesuhe või mitte. Nende vektorite põhjal õpib mudel hiljem ise väljundit ennustama.

(4) *Mari* kirjeldas õhinal, kuidas *ta* väiksena „*Sipsikut*“ luges: „*Ma* ei suutnud *seda raamatut* kohe kuidagi *käest* ära panna!“

Näites (4) on kindel üksik viitesuhe *Sipsikut* ja *seda raamatut* vahel. Sõna *käest* ei viita kuhugi. Kui mudel määrab osutusele aga mitu viitealust (näiteks sõnale *Ma* on õiged nii *ta* kui ka *Mari*) ehk leiab mitu paari, milles üks osutus kordub, tuleb valida sobivaim. Selleks kasutatakse klasterdamist (Novák 2018: 24). Neid on kahte sorti: *lähim-enne klasterdus* (ing *closest-first clustering*) (Soon jt 2001) eelistab asendussõnale lähimat kandidaati ja *parim-enne klasterdus* (ingl *best-first clustering*) (Ng, Cardie 2002) eelistab mudelilt kõige kõrgema tõenäosuse skoori saanud kandidaati. Taoline klasterdamine aitab välja sorteerida ka vale viitesuhte-sildi saanud paare.

Osutus-paari mudeli suureks miinuseks on asjaolu, et see vaatab paare eraldi ega arvesta teiste kandidaatidega. Kahe sõna kesksus limiteerib aga algoritmile antava teabe ehk tunnuste hulka. **Osutus-järjestus mudel** (ingl *mention-ranking* model) vaatleb viitealuste kandidaate (kõiki osutusi) üheaegselt ning järjestab need. See tähendab, et

parima viitealuse kandidaadi valimine käib, erinevalt osutus-paari mudelist, mis hindab iga paari eraldi, juba mudeli sees. See annab juurde võrdlusmomendi teiste kandidaatidega. Selle lähenemise miinuseks on see, et mudel ei saa asendussõnale viitealust määramata jätta ja mitte-viitelised asendussõnad tuleb enne mudeli kasutust välja sorteerida. Sellegipoolest on osutus-järjestus mudelid saanud osutus-paari mudelitest paremaid tulemusi. (Ng 2017: 3, Novák 2018: 25–26)

Mõlemad osutus-põhised mudelid on populaarsed oma kiiruse ja lihtsuse tõttu (Novák 2018: 26), kuid nende puuduseks on vähene väljenduslikkus (ingl *expressiveness*). Kuna mudeli sisendiks on vaid kahe osutuse tunnused ja nende võimaliku viitesuhte info, ei pruugi sellest piisata, et valed suhted välja sorteerida. (Ng 2017: 3, Novák 2018: 25) See tähendab, tunnused ei väljenda piisavalt infot konteksti ja viitesuhte kohta ja on liiga vähe väljenduslikud. Seda probleemi leevendab **olemi-põhine mudel** (ingl *entity-based model*), mis tegutseb klastritasandil[6], otsides paaride asemel osutusklastreid, milles olevad osutused viitavad kõik ühele olemile. Lisaväärtuse annavad just klastriülesed tunnused kõikidelt osutustelt selles klastris. (Ng 2017: 3) Teisisõnu, otsitakse pigem viiteahelaid, kui üksikuid viitesuhteid ja enam ei piirduta paaridevaheliste tunnustega, vaid saab vaadelda kogu klastri tunnuseid. Näite (4) osutused klasterdataks olemi-põhise mudeliga kolme gruppi: esimeses oleks osutused *Mari, ta, Ma,* teises osutused *Sipsikut, seda raamatut,* ja kolmandas osutus *käest*.

Olemi-põhist mudelit saab edasi arendada **jaotuspõhiseks mudeliks** (ingl *partition-based model*). See mudel otsib samuti klastreid, milles on kõik ühele olemile viitavad osutused, kuid meetod selleks on teine. Mudel saab ette erinevad suvaliselt genereeritud kandidaatide jaotused (klastrid) ning valib välja kõige tõenäolisema, mitte ei loo ise uut klastrit. (Ng 2017: 3–4) Tulemus on olemi-põhise mudeliga sama: olemiklastrid.

Viitesuhte automaatsele lahendamisele võib läheneda ka **graafi-põhiselt** (ingl *graph-based*) (Ng 2017: 4). Graafi-põhisel lähenemisel vaadeldakse osutusi kui tippe graafis, vaja on leida vaid õiged harud. Igale tipuvahelisele suhtele ehk harule arvutatakse kaal. Kaalu arvutusmeetod sõltub konkreetsest rakendusest, näiteks on

[6] Siin võib klastrist mõelda kui osutuste kogumist või viiteahelast.

kasutatud osutus-paari mudelit ja statistikat. Seejärel jaotab jaotusalgoritm tipud paarideks (ehk viitesuheteks) või gruppideks (ehk viiteahelateks). Selle meetodi tugevus on samuti asjaolu, et tipud on omavahel ühendatud ja niimoodi saab kätte rohkem tunnuseid. See vähendab konteksti ja info vähesuse tõttu tekkinud automaatselt valesti määratud viitesuhteid ja -ahelaid. (Sapena jt 2013: 853–854)

Fernandes jt (2012) lõid **puul põhineva mudeli** (ingl *tree-based model*), mille väljund on viitesuhte puu (ingl *coreference tree*). See puu on, nagu graafki, terve dokumendi kohta. Graafil ja puul põhinevate mudelite erinevusena saab välja tuua selle, et puu struktuur on spetsiifiline ega sarnane graafile: puul põhineval mudelil on osutused eri tasanditel. Puu loomiseks kasutatakse pertseptroni (ingl *perceptron*), mitte kaale iga osutuse vahel nagu graafi-põhine mudel. Põhimõtteliselt sarnaneb see natukene jaotuspõhise ja graafil põhineva mudeliga, kuna hiljem tükeldatakse puu väiksemateks osadeks, klastriteks. Selle mudeli plussiks on jällegi suurem tunnuste hulk, st arvestatakse kõikide osutuste tunnustega.

Kõigi seni mainitud mudelitüüpide koostamisel on läbivaks jooneks alati olnud küsimused „Millised tunnused peaksid olema sisendvektorites? Kas mudel saab piisavalt palju infot otsustamiseks? Mis tunnused on vektoritest puudu, millised üle?“. Vastuseid nendele küsimustele limiteerivad võimalused sobivaid tunnuseid automaatselt kätte saada. Mängida saab ikka nende tunnustega, mis on kättesaadavad.

Durrett ja Klein (2013) kasutasid tunnuste-defitsiidi leevendamiseks teistmoodi lähenemist: viitealuse struktuuril põhinevat mudelit (ingl *antecdent-structure based model*). **Viitealuse struktuuril põhinev mudel** modelleerib lihtsate tunnuste abil ise keerulisemaid keeleteaduslikke mustreid, mida niisama on automaatselt keeruline leida ja koostada. Seetõttu ei pea tegelema tunnuste disainimisega ega (näiteks reeglipõhise) eelfiltreerimisega. Arvutuslikult on see mudel optimaalne, kuna sisendvektorite loomisele ei kulu palju ressurssi ja latentsete (ehk peidetud, otseselt mitte kättesaadavate) tunnuste tuvastamisega tegeleb mudel ise. Ka algoritmiliselt on see mudel silmatorkav. Algoritmi väljund ei ole klass, viitealus ega klaster, vaid vektor, kus on iga dokumendi osutuse kohta käiv viitealus või viitealused (või null, kui osutus ei viita kuhugi). Seega tuvastab mudel kõik viitesuhted korraga ega vaja järeltöötlust

(jaotamist vms). Durrett ja Klein arutlevad oma artiklis palju ka semantilise infoga tunnuste lisamise üle, kuid nemad sellega märkimisväärset kvaliteeditõusu tulemustes ei saavutanud.

Maailmateadmuse implementeerimine loomuliku keele töötluse vahenditesse on siiani keeruline ja mahukas lahendamata ülesanne. Lugejale võib näidetes (5) ja (6) üsna ilmselge olla, mis on viitealus. Viitesuhte lahendaja jaoks on mõlemas näites *Bill* ja *John* võrdsed kandidaadid. Kui eelistada süntaktilist ühildumist, määrataks mõlemas näites viitealuseks *John*. Kui eelistada lähimat nimisõna, siis *Bill*. Kui tahta õigesti määrata, peab lahendaja teadma, et purjus inimesed ei tohi sõita ja neilt võetakse autovõtmed ära. Kõigele sellele mõeldes tuleb meeles pidada, et tarbijateni jõudvate süsteemide puhul on oluline ka kiirus ja mudeli optimaalsus, mida seni semantilised analüüsid tagada ei suuda. Durrett ja Klein peavad masinale arusaadaval ja keeletehnoloogidele kasutataval kujul semantilise info modelleerimist ikka veel „lahinguks ülesmäge“ (Durrett ja Klein 2013: 1978).

(5) John peitis *Billi* võtmed ära. *Ta* oli purjus. (Lappin 2005)
(6) *John* peitis Billi võtmed ära. *Ta* mängis talle vingerpussi. (Lappin 2005)

Viitesuhete automaatne lahendamine ei ole kompleksne ülesanne mitte ainult piisava (ja korrektse!) vajalikku keeleteadusliku info kättesaamise ja mudelile sobivale kujule kodeerimise tõttu. Oluline on tegeleda ka nende osutustega, mis ei kuulu ühtegi ketti, st ei viita kuhugi ega oma viitealust. Paljud (aga mitte kõik) osutus-põhised algoritmid määravad kõikidele osutustele meelevaldselt mingisuguse viitesuhte. Siin tuleb appi **ühismudel** (ingl *joint model*), mis tegelikult koosneb mitmest järjestikusest mudelist. Iga mudel tegeleb spetsiifilise ülesandega ning annab oma väljundi järgmisele mudelile. Näiteks määrab kõigepealt ära asendussõna viitesuhtelisuse ja seejärel leiab viitesuhtelistele asendussõnadele viitealused. Ühismudelite abil saab lahendamist teha ka mitmeetapiliselt: näiteks kasutada esimest mudelit tunnuste ning kaalude arvutamiseks-loomiseks, mis hiljem lähevad teise, lahendaja-mudeli sisendiks (Song jt 2012). (Ng 2017: 4)

Ühismudeliga on veidi sarnane **kergem-enne mudel** (ingl *easy-first model*), mis kasutab samuti mitut mudelit, kuid eelnevad mudelid ei tooda lõpu-lahendajale

tunnuseid ja parameetreid, vaid teevad juba otsuseid viitesuhete kohta. Esimesed mudelid lahendavad lihtsamad juhud, jättes raskemad osad järgmisele. Seda võib vaadelda kui sõelasüsteemi, milles on reeglite asemel mudelid. (Ng 2017: 4)

Kõik ülaltoodud näited sõltuvad märgendatud korpustest. Seega sõltub mudeli tulemuste kvaliteet ka korpuse kvaliteedist (andmete õigsusest ja viitesuhete varieeruvusest) ja suurusest. **Pool-juhendatud** (ingl *semi*-supervised) ja **juhendamata** (ingl *unsupervised*) **masinõppe mudelid** tegelevad märgendamata andmestiku pealt õppimisega. Pool-juhendatud ja juhendamata algoritme rakendatakse erinevates masinõppe valdkondades, eriti masintõlkes. See tuleb kasuks ka viitesuhete lahendamisel (Ng 2017: 4), kui märgendatud andmestik puudub või pole seda piisavalt palju. Näiteks Charniak ja Elsner (2009) lõid pronominaalsete viitesuhete lahendaja *Expectation Maximization* algoritmiga, mis õppis selgeks pea kõigi vajalike sisendtunnuste genereerimise lisaks viitesuhete lahendamisele.

Loomuliku keele töötluses on läbi aja erinevatele ülesannetele lähenetud alguses reeglipõhiselt, seejärel masinõppepõhiselt. Nüüd on populaarne rakendada närvivõrke. **Närvivõrgud** (ingl *neural networks*) on masinõppe alamliik, mis põhineb elusolendi närvisüsteemi ülesehituse ideel. Närvivõrkudest võib mõelda kui masinõppe algoritmist, millega saab lahendada samu ülesandeid nagu ülalpool toodud on. Esimesed katsetused viitesuhete lahendamisel närvivõrkudega on paljulubavad: mittelineaarsed mudelid saavutasid masinõppe mudelitega sama häid tulemusi (Ng 2017: 5).

Närvivõrgud on praegu keeletehnoloogias populaarne uuendus, mida pidevalt arendatakse ja millest seni hüppeliselt tõhusamat alternatiivi leitud pole. Kogu moodsa masinõppe tuhinas ei ole reeglipõhised lähenemised siiski unustuse hõlma vajunud (vt nt Lee, H. jt 2017: 2, 6–7): neid kasutatakse tihti osutuste eelfiltreerimiseks või muudes taolistes kohtades viitesuhete lahendamise voos, kus lingvistilised teadmised ja reeglid aitavad vähendada mudeli tööd (või viitealuste kandidaatide hulka).

Värskeima kompaktse ülevaate viitesuhete lahendamisest (eriti inglise keele põhiselt) ja viimase aja arengutest mudelipõhiste lähenemiste seas leiab Jurafsky ja Martini õpiku

„Speech and Language Processing“ 22. peatüki „Coreference Resolution“ mustandist[7] (2019). Järgnev alapeatükk annab põgusa ülevaate närvivõrkudest viitesuhete lahendamise kontekstis.

1.4. Närvivõrgud asendussõnade automaatse lahendamise kontekstis

Viimase kahe aasta jooksul on viitesuhete lahendamisele inglise keeles üha rohkem tähelepanu pööratud ning uue tulijana on katsetatud just närvivõrgu meetodeid. Väga hea värske ülevaate nendest annavad Stylianou ja Vlahavas oma artiklis „A Neural Entity Coreference Resolution Review“ (2019), mida tasub detailsema ülevaate saamiseks lugeda. Antud alapeatükk ei sea endale eesmärgiks välja tuua kõik viitesuhete närvivõrk-lahendajad, vaid kirjeldab põgusalt mõnda kirjanduses enimmainitut.

Kenton Lee jt (2017) muutsid viitesuhete lahendamise mõtteviisi. Kui enne hoiti osutuste eraldamist ning nendevaheliste viitesuhete leidmist lahus, siis nemad lükkasid need oma mudeliga kokku. Nad lõid inglise keelele osutus-järjestus närvivõrgu, millel puudub osutuste eraldamise etapp: mudel saab sisendiks tekstist leitud kõikvõimalikud mitmikud (teisisõnu n-grammid, maksimum on 10-gramm) ehk lõigud tekstist (ingl *span*). Iga mitmiku puhul hindab mudel, kas tegu on üldsegi osutusega ja arvutab välja viitesuhte skoori mitmiku ja kõikide talle eelnevate mitmikute vahel. Mitmikute hulka on lisatud ka fiktiivne tühi osutus (ingl *dummy token*), mis saab kõrgeima skoori viitesuhtes siis, kui tegu pole olemi osutusega või kui see alustab uut viiteahelat (talle ei eelne viitealust, aga ta on mõne järgneva osutuse viitealus). Selle lähenemise miinuseks on suur algoritmiline keerukus ($O(T^4)$, kus T on dokumendi pikkus), mida üritatakse vähendada aktiivse madala osutus-skooridega mitmikute väljaviskamise ning viitealuse kandidaatide hulga piiramisega kauguse alusel. Viimane aga jätab automaatselt välja loomulikus keeles esinevad kaugele viitavad (pika vahemikuga) viitesuhted. (Kenton Lee jt 2017: 4, Lee, K. jt 2018: 1, 3–4)

Hiljem arendas väiksem osa eelmise mudeli loojate seast seda mudelit edasi (Lee, K. jt 2018), muutes mudeli arhitektuuri iteratiivseks. Igal iteratsioonil täpsustatakse olemasolevaid mitmikute (seekord on maksimumiks 30-gramm) esitusi ehk

[7] Õpik ilmub tõenäoliselt 2020. aasta sees.

tunnusvektoreid (ingl *span representations*) eelmise iteratsiooni alusel. St, osutuste esitusi ei genereerita korraga, vaid sammhaaval. Samuti tehti viitealuse otsingud kaheetapiliseks: alguses rakendatakse ebatäpsemat võimalike viitealuste sõela, mis jätab alles tõenäolisemad viitealuse kandidaadid (kauguse alusel kandidaatide elimineerimine kadus). Alles seejärel rakendatakse leitule algoritmiliselt kulukamat järjestusfunktsiooni (ingl *scoring function*) õige viitealuse leidmiseks. (Lee, K. jt 2018: 1) Selle uue mudeli tulemused on palju stabiilsemad ning kõrgemad (mitmiku maksimumpikkusest sõltumata) võrreldes eelmise mudeliga (Lee, K. jt 2018: 3).

Kahe mudeli õiglasema võrdlemise tarbeks muutsid Lee, K. jt (2018) esimese mudeli üht olulist (semantikat edastada üritavat) tunnust, sõnavektorit (ingl *embedding*), ajakohasemaks. Kui esimene mudel kasutas GloVe-d (Pennington jt 2014) ja Turiani jt sõnavektoreid (2010), siis teine mudel kasutas juba ELMo-t, mis tegi oma debüüdi (Peters jt 2018) teise mudeliga samal aastal. Joshi jt (2019) proovisid sama mudelit hoopis BERT-iga (Devlin jt 2019) ja võrdlesid seda ELMo variandiga. Mõnes olukorras oli BERT parem, kuid silmapaistvaid kvaliteedierinevusi ELMo ja BERT-i vahel ei leitud.

Taoline võidujooks sõnavektorite arenguga illustreerib, kuidas inglise keelele on tõhusa viitesuhete lahendaja üheks suurimaks takistuseks semantika pädev esitus masinõppes. See tähendab, et kõik muud (süntaktilised ja morfoloogilised) tunnused on läbi töötatud ning puudu on vaid keeletehnoloogias üldine lahendamata probleem: semantika tõhus esitus ja käsitlus.

Teiseks suurimaks takistuseks on närvivõrkude piiratus. Lisaks semantika esitusviisidele arenevad ka närvivõrkude algoritmid, mis parendavad pidevalt seniseid tulemusi. Subramanian ja Roth (2019) juhtisid tähelepanu seni treenitud närvivõrkude madalale üldistusvõimele (ingl *generalization*), st võimele töötada oleminimedega, mida treenimisel ei nähtud. Nad testisid Lee, K. jt mudeleid sellise andmestiku põhjal, kus treening- ja testandmestikus ei kattunud ükski inimese ega asukoha nimi. See vähendas märgatavalt närvivõrgu täpsust. Seejärel mugavdasid nad Lee, K. jt teist mudelit Miyato jt poolt välja töötatud FGSM meetodiga (ingl *adversarial fast-gradient-sign-method*) (Miyato jt 2017), mis põhimõtteliselt tegelebki närvivõrkude üldistusvõime parandamisega. Saadud närvivõrguga edastasid nad kõigi eelkäijate tulemused ja sellest sai tolle hetke *state-of-the-art*.

Eesti keelele pole seni katsetatud veel ühtegi närvivõrkudel põhinevat viitesuhete lahendajat, semantikaga või ilma. Seda puudujääki proovibki antud töö likvideerida.

2. MATERJAL JA TREENINGANDMESTIKU LOOMINE

2.1. Asendussõnade suhtes käsitsi märgendatud korpus viitesuhete lahendamise kontekstis

Projekti „Sihipärane süntaks korpuse jaoks“ (2015–2017) raames loodi ca 107000 tekstisõna suurune asendussõnade suhtes käsitsi märgendatud ajalehetekstide korpus. Selle korpuse märgendamisreeglite ja -probleemide kohta saab täpsemalt lugeda Freienthali bakalaureusetöö peatükist 2.1 (Freienthal 2018: 18–23). Hiljem täiendati seda korpust aastatel 2018–2019 Haridus ja Teadusministeeriumi keeletehnoloogia teadus- ja arendustegevuse programmi „Eesti keeletehnoloogia 2018-2027“ projekti „Eesti keele universaalse süntaksi vahendid ja rakendused” raames veel ca 147000 sõne võrra, eesmärgiga luua masinõppele suuremat andmestikku. Iga teksti märgendas kaks inimest eraldi, hiljem vaatas üks neist erinevused üle ja ühtlustas ära. Peamine märgendaja oli töö autor.

Korpus koosneb ajalehetekstidest: selles on artikleid 2001. ja 1999. aasta Eesti Ekspressist, 2006. ja 2007. aasta Eesti Päevalehest, 2002. aasta Maalehest, ajalehest Luup (väljavõtteid aastatest 2000, 2001 ja 2002), 2000. ja 1998. aasta Postimehest. Ajalehtede seast eristub teadusajakiri Eesti Arst (aastast 2004).

Korpuses on viitesuhte suhtes käsitsi märgendatud järgnevad pronoomenid:

- näitav asesõna *see* ja *need*;
- küsiv-siduvad asesõnad *kes* ja *mis*;
- isikulised asesõnad *mina, sina, tema, meie, teie* ja *nemad*.

Ära märgiti nii katafoorsed kui ka anafoorsed viitesuhted. See tähendab, et viitealus võib asendussõnast paikneda ees- või tagapool. Viitesuhted võivad olla nii lausesisesed kui ka lauseteülesed. Kokku on korpuses 8323 viitesuhtega pronoomenit, nendest 482-l on mitu viitealust.

Korpus seab tööle ka teatud piirangud: selles puuduvad viiteahelad, viitealuseks võib olla ainult üks sõna ning märgendatud on vaid pronominaalsed viitesuhted. Viiteahelate puudumine tähendab seda, et klasterdamismeetodeid ei ole mõttekas kasutada, kuna ühel asendussõnal on üks ühe tähendusega viitealus (ehk ei moodustu olemiklastreid).

Tekstis, kus ühele ja samale olemile viidatakse mitu korda erinevate sõnadega, valiti käsitsi märgendades viitealuseks pronominaalsele asendussõnale ainult üks, asendussõnale lähim ja tähenduslikult täpseim viitealus.

Kaks või enam viitealust võib asendussõnal olla juhul, kui tegu on mitmusliku sisuga asendussõnaga nagu *meie* või *nemad* ja nende kogutähenduse märkimiseks on vaja viidata mitmele tähenduslikult erinevale sõnale. Näites (7) ongi asendussõna *need* kogutähendus loetelu *Piiripost, piirivalvur* ja *piirikoer*. Vaid ühele neist viitamine ei anna kogu tähendust edasi.

(7) *Piiripost, piirivalvur* ja *piirikoer* - *need* on lahutamatud sümbolid.

Eesti korpuses on viitealuseks märgitud vaid üks sõna ka siis, kui tegelik viitealus on terve fraas (sel juhul on viitealus antud fraasi põhi). Seda otsust mõjutasid korpuse „Eesti keele sõltuvuspuude panga (EDT)“ formaadi piirangud (vt lähemalt (Muischnek, Müürisep 2016). Eelmises peatükis mainitud tööd tegelevad peamiselt nimisõnafraasiliste osutustega, see tähendab, et treeningmaterjalis on viitealuseks märgitud terved fraasid. See mõjutab kogu lahendamisprotsessi: asendussõnade automaatse lahendamise eeltöötluses peab osutuste eraldamise meetod ja fookus olema teistsugune ning eesti keele lahendaja mudel ei saa kasutada mitmeid populaarseid tunnuseid muude keelte lahendajates nagu osutuse esimene sõna ja osutuse viimane sõna, sest osutus ongi üks sõna.

Eelmises peatükis toodud mudelite näited keskendusid peamiselt kõikide viitesuhete lahendamisele, otsides samale olemile viitavaid sõnade ja fraaside klastreid. Korpuses on märgendatud vaid pronoomenid ja seetõttu kõigi viitesuhete lahendamisele keskenduda ei saa. Pronominaalsete viitesuhete automaatne lahendamine on kõikide viitesuhete lahendamise keerulisem alamülesanne (Zhang jt 2019). Keeruliseks teeb selle asesõnade kontekstitundlikkus: kui samale olemile viitavate nimisõnade üheks häid tulemusi andvaks tunnuseks on sõnade osaline või täielik kokkulangevus (nt *president Kaljulaid* langeb osaliselt kokku fraasiga *Kersti Kaljulaid*), siis pronoomenite puhul seda kasutada ei saa. Nende lahendamiseks tuleb tugineda muudele tunnustele. Samas kaob leheküljel 16 mainitud üldistusvõimetus, kuna lemmade võrdlemine osutus-pronoomen paari puhul pole mõistlik.

2.2. Asendussõnade suhtes käsitsi märgendatud korpuse eeltöötlus

Tartu Ülikooli keeletehnoloogia teadur Eduard Barbu ning keeletehnoloogia vanemteadur ja arvutilingvistika dotsent Kadri Muischnek tegelesid töö autori abiga korpuse eeltöötlusega, tunnuste eraldamisega ning töövoo koostamisega. Viimase kahe juures oli töö autoril ning Muischnekil pigem nõustav roll.

Enne treeningmaterjali korpusest eraldamist teisendati korpus eesti keele sõltuvuspuude panga formaadilt (EDT) *Universal Dependencies* (UD) formaadile CONLL kujul ja puhastati korpus ebasobivatest viitesuhetest. Välja jäid viitesuhted, kus pronoomen on märgitud määratlejaks nagu *seda* näites (8). Alles jäeti ainult sellised viitesuhted, kus asendussõna on pronoomen ning viitealus on nimisõna, pärisnimi või pronoomen. Viitealus võib olla pronoomen näiteks olukorras (9), kus asendussõna *Me* viitealuseks on loetelu m*ina* ja *koeraga*. Taoline filtreerimine kitsendas ülesannet ja vähendas oluliselt viitealuste kandidaatide hulka. Välja jäid näiteks sellised märgendused, kus viitealus on adjektiiv (näide (10)) või adverb (näide (11)). Adverb võib korpuses olla viitealusena koopulalausetes, kus algselt oli viitealuseks märgitud osalause juurtipuks olev *olema*-verb (näites (11) verb *pole*), mille kaudu viidatakse kogu kõrvallausele. UD formaadis pole koopulalausetes juurtipuks öeldis, vaid on mõni muu lauseliige, ja seetõttu muutus teisendamisel ka viitealus. Eesti keele koopulalausetest UD formaadis saab täpsemalt lugeda Muischneki ja Müürisepa artiklist (2017). Kuna antud töös verbe viitealuste kandidaatidena ei vaadata, ei mõjuta verbide ja adverbidega viitesuhete välja heitmine töö tulemust.

(8) Et oma *territooriumi* märgistada, kasutame *seda* kohta korduvalt, jätame sinna oma esemeid, asetame piirdeid, väldime puudutusi, pilke.

(9) Eile läksin *mina koeraga* jalutama. *Me* nägime ülihead välja.

(10) *Halvim, mida* Vello suudab kellegi kohta öelda, on „noh, ta on ju muidu kena inimene“.

(11) Meil pole nii *palju* prantsuse ja inglise keele oskusega kohtunikke, kes hakkaks meie asju kaitsma. *See* on järgmine argument kiire liitumise vastu.

Samuti eemaldati veel mõned EDT formaadist UD formaati teisendamisel tekkinud vead.

Ajaleheartiklid olid enne märgendamist jagatud märgendamiskeskkonda mahupõhiselt, mis tähendas seda, et osad artiklid lõigati keskelt pooleks ja lõpp pandi teise faili. Kahe faili vahelisi viiteid aga bratis märgendada ei saanud ja nii jäi päris mitu võimalikku viitesuhet märgendamata. See viga parandati samuti enne tunnuste eraldamist ära. Taolise eeltöötluse tulemusena jäi korpusesse alles 6866 viitesuhet, millest 289 on mitme viitealusega.

Kuna suur osa asendussõnade *see* ja *need* viitealustest on verbid (40% suurendamata korpuse viitesuhetest (Freienthal 2018: 32–33)) ja verb-viitealuselised viitesuhted eemaldati korpusest, siis nende automaatsele lahendamisele esialgu ei keskendutud. Asendussõnad, millele käesolevas töös automaatset lahendajat luuakse, on järgnevad:

- küsiv-siduvad asesõnad *kes* ja *mis* ja
- isikulised asesõnad *mina, sina, tema, meie, teie* ja *nemad*.

2.3. Treeningandmestiku loomine ehk tunnuste eraldamine

Töö järgmises etapis tegeleti tunnuste eraldamisega ehk masinõppe algoritmidele treeningmaterjali koostamisega. Selleks lõi Eduard Barbu veel avaldamata arendusaluse (ingl *baseline*), mis eraldab tekstidest võimalike viitesuhete tunnuste vektoreid, mille seast mudelid päris viitesuhted välja sorteerivad.

Kuna töö kasutab osutus-paari mudeli põhist lähenemist (vt lk 10), loodi korpuse põhjal viitesuhete vektorid, milles olevad tunnused kirjeldasid selles viitesuhtes olevat asendussõna, viitealust ja nendevahelisi tunnuseid. Viitealuseid ühele asendussõnale ei ole mõistlik otsida kõigi osutuste seast tervest tekstist. Seetõttu piirati viitealuste otsimisvahemikku Freienthali bakalaureusetöö (2018) ning katsete põhjal (vaatasime kui palju viitesuhteid nende reeglitega korpusest välja jääb). Parimateks vahemikeks osutusid asesõnade *mina* ja *meie* puhul sama lause kõik osutused (ka katafoorsed ehk asendussõnale järgnevad viitealuste kandidaadid) ja kuni kolm lauset eespoolt. Asesõnade *sina* ja *teie* puhul otsitakse viitealuste kandidaate samuti kuni kolm lauset eespoolt, kuid samast lausest vaadatakse vaid asendussõnale eelnevaid osutusi. Asesõnade *kes* ja *mis* puhul otsitakse viitealuste kandidaate samast lausest neile eelnevate osutuste seast. Viimased on oma olemuselt ka kõige lihtsamad relatiivlausete

laiendajad (vt näide (10), kus *mida Vello suudab kellegi kohta öelda* on relatiivlause ja *mida* on selle laiendaja).

Negatiivseid näiteid treeningmaterjali võeti samade reeglite alusel. Kõik nimisõnad, pärisnimed ja pronoomenid, mis mingi asendussõna määratud viitamisvahemikku sattusid, kuid polnud selle asendussõna viitealused, pandi paari antud asendussõnaga viitesuhte puudumise näitena.

Eduard Barbu eraldas korpusest Kadri Muischneki ja töö autori abiga 32 viitesuhete tunnust, mille põhjal koostati treeningmaterjal. Need on järgnevad (toodud vektoris esinemise järjekorras):

1. kaugus lausetes (0 – asendussõna ja viitealus on samas lauses, 1 – eelnevas lauses jne),
2. ühildumine arvus (0 – ei, 1 – jah),
3. viitealus on nimisõna (0 – ei, 1 – jah),
4. viitealus on pärisnimi (0 – ei, 1 – jah),
5. viitealuse kääne (kategooriline tunnus teisendatud arvuks vahemikus [0,14]),
6. pronoomeni kääne,
7. viitealuse asukoht (0 – lause esimene sõna , 1 – lause keskel, 2 – lause viimane sõna),
8. nii viitealus kui ka kääne on subjektid (0 – ei, 1 – jah),
9. viitealusele eelneva sõna sõnaliik (ingl *part-of-speech*) (kategooriline tunnus teisendatud arvuks vahemikus [0, 16]),
10. viitealusele eelnevale sõnale eelneva sõna sõnaliik,
11. viitealusele järgneva sõna sõnaliik,
12. viitealusele järgnevale sõnale järgneva sõna sõnaliik,
13. asendussõnale eelneva sõna sõnaliik,
14. asendussõnale eelnevale sõnale eelneva sõna sõnaliik,
15. asendussõnale järgneva sõna sõnaliik,
16. asendussõnale järgnevale sõnale järgneva sõna sõnaliik,
17. viitealuse süntaktilise ülemuse sõnaliik,
18. asendussõna süntaktilise ülema sõnaliik,
19. viitealuse süntaktiline funktsioon (kategooriline tunnus teisendatud arvuks

vahemikus [0, 44]),

20. asendussõna süntaktiline funktsioon,
21. asendussõnaga lause pikkus sõnedes,
22. viitealuse sõnaliik,
23. asendussõna sõnaliik,
24. asendussõna lemma on *kes* (0 – ei, 1 – jah),
25. asendussõna lemma on *mis* (0 – ei, 1 – jah),
26. normaliseeritud asendussõna kaugus viitealusest,
27. viitealuse sagedus kuni 10 lauset eespool olevas tekstis,
28. asendussõna kaugus viitealusest sõnedes,
29. asendussõna koosinus-sarnasus viitealusega,
30. sõnavektor puudub ja koosinus-sarnasust arvutada ei saa (0 – ei, 1 – jah),
31. viitealuse sagedus kogu tekstis,
32. viitealuse abstraktsuse skoor,
33. kategooria ehk silt ehk see binaarne väärtus, mida ennustama hakatakse (0 – ei ole viitesuhtes, 1 – on viitesuhtes).

Asendussõna koosinus-sarnasus viitealusega (tunnus 29) arvutati Eesti Keeleressursside Keskuselt saadud sõnavektorite abil (Entu). Sõnavektori puudumist või olemasolu näitab tunnus 30. Viitealuse sagedus kogu tekstis (tunnus 31) võeti antud korpusefaili üleselt. Kuna failides ei ole artiklite vahesid märgendatud, võis juhtuda, et sageduse arvutamisel otsiti osutust ka teiste artiklite seest. Abstraktsuse skoor (tunnus 32) võeti Eleri Aedmaa doktoritöö „Detecting compositionality of Estonian particle verbs with statistical and linguistic methods“ (2019) käigus valminud leksikonist (Abstractness_ET). Kui leksikonist vastavat viitealust ei leitud, pandi skooriks 0. Kahe või enama viitealusega asendussõnade puhul (vt näidet (7)) loodi igale viitealusele eraldi tunnuste vektor selle asendussõnaga.

Ülal toodud viisil tunnuseid eraldades jõudis treeningmaterjali korpusesse väga tasakaalust väljas materjal: 6230 positiivset näidet (viitesuhtes sõnapaari) ja 155198 negatiivset näidet (viitesuhteta sõnapaari). See teeb 25 negatiivset näidet ühe positiivse näite kohta.

3. NÄRVIVÕRGUD JA NENDE ANALÜÜS

Korpusest eraldatud materjali põhjal treeniti osutus-paari mudelil (vt lk 10) põhinevad närvivõrgud. Parimate närvivõrgu parameetrite leidmisele ja katsetamisele kulus mitu kuud, mille käigus läbis autor Andrew Ng kursuse „Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization“ videoloengud (Ng coursera kursus) ja õppis iseseisvalt juurde. Autor tänab Tartu Ülikooli keeletehnoloogia nooremteadurit Lisa Yankovskayat, kes tutvustas autorile MCC-d, vihjas, millistele parameetritele võiks veel tähelepanu pöörata, ning vastas teoreetilistele mureküsimustele.

Parimaid saadud närvivõrke nimetatakse siin NN1-ks ja NNa-ks. Kõikide selles töös mainitud närvivõrkude ja treenimisandmestikuga saab lähemalt tutvuda aadressil https://github.com/Lindafr/EstPronCorefNN. **NN1** ja **NNa** on sama ülesehitusega viiekihilised laiad närvivõrgud (vt joonist 1 lk 27). Mõlemas närvivõrgus kasutatakse reguleerimiseks **väljajätumeetodit** (ingl *dropout,* (Hinton jt 2012)) ja teises kihis **L2-regulariseerimist** (ingl *L2 regularization*, vt nt (Goodfellow jt 2016: 227–230)) ning närvivõrgu kiiremaks ja kergemaks treenimiseks pea igas kihis **ploki normaliseerimist** (ingl *batch normalization,* (Ioffe ja Szegedy 2015)). Kihte aktiveerib **ELU** (Clevert jt 2016). Vaid väljundkihti aktiveerib binaarsele klassifikatsioonile sobiv **sigmoidfunktsioon** (ingl *sigmoid function,* vt nt (Goodfellow jt 2016: 65–66)), mis väljastab arvu vahemikus (0,1). Seda arvu võib võtta kui tõenäosust, et paar on viitesuhtes. Kahe närvivõrgu ainuke erinevus on nende optimeerijas: NN1-e optimeerib õpisammuga 0,001 **Adam** (Kingma, Ba 2017), NNa-d optimeerib õpisammuga 0,002 **Adagrad** (Duchi jt 2011).

Selles peatükis võrreldakse nende närvivõrkude tulemusi omavahel. Seejärel treenitakse neid närvivõrke erinevate parameetrimuudatustega uuesti, et võrrelda andmestiku kodeerimise, treening- ja valideerimisandmestiku positiivsete ja negatiivsete paaride osakaalu, epohhide (ingl *epoch*), õpisammu (ingl *learning rate*) ja ploki suuruse (ingl *batch size*) mõju närvivõrgu tulemustele.

Peatükk algab tulemuste analüüsimiseks oluliste mõõdikute lahti seletamisega ja lõpeb sama ülesannet lahendavate teiste masinõppe meetodite ja närvivõrkude võrdlemisega ja analüüsi tulemuste kokkuvõttega.

3.1. Närvivõrkude tulemuste hindamise mõõdikud

Viitesuhete lahendamise hindamiseks on loodud mitmeid parameetreid nagu MUC-F, CEAF, B-cubed ja BLANC (vt täpsemalt Luo ja Pradhan 2016). Need parameetrid on loodud eelkõige pidades silmas viiteahelate lahendamist. Kuna töö tegeleb ahelate asemel viitesuhete lahendamisega, ei ole need mõõdikud antud töö kontekstis aktuaalsed. Siin töös analüüsitakse tulemusi segadusmaatriksi, F1, MCC, õigsuse, (vt nt (Chicco, Jurman 2020)) I ja II tüübi vea (vt nt (Banerjee jt 2009)) ja kahju (vt nt (Godoy 2019)) põhjal.

Segadusmaatriks (ingl *confusion matrix*) (tabel 1) annab ülevaate, kui palju tõeselt positiivseid (TP), valepositiivseid (FP), tõeselt negatiivseid (TN) ja valenegatiivseid (FN) väärtuseid antud mudel testandmestikul annab. Selle põhjal saab hinnata mudeli edukust kummaski rühmas (viitesuhtes ja viitesuhteta). Segadusmaatriksi põhjal saab arvutada I ja II tüübi viga. Nullhüpoteesi, et antud paar on viitesuhtes, põhjal näitab **I tüübi viga** (ingl *type I error*), kui palju ennustatult viitesuhtes paaridest on tegelikult valepositiivsed. **II tüübi viga** (ingl *type II error*) näitab, kui palju ennustatult viitesuhteta paaridest tegelikult on viitesuhtes.

Tabel 1. **Segadusmaatriks.**

		Mudeli ennustus	
		Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes	TP	FN
	Viitesuhteta	FP	TN

a) $I\ tüübi\ viga\ = \frac{\mathrm{FP}}{\mathrm{TP+FP}}$

b) $II\ tüübi\ viga\ = \frac{\mathrm{FN}}{\mathrm{TN+FN}}$

F1-st võib mõelda kui saagise (ingl *recall*) ja täpsuse (ingl *precision*) harmoonilisest keskmisest. **Täpsus** ehk positiivne ennustusväärtus näitab, kui palju positiivseks

ennustatud paaridest ennustati õigesti. **Saagis** ehk tõeselt positiivsete määr[8] näitab, kui palju viitesuhtes paaridest ennustati õigesti. Pythoni masinõppe mooduli Scikit-learn'i (Pedregosa jt 2011) alt leiab funktsiooni *classification_report,* mis paneb täpsuse, saagise ja F1 ühte tabelisse. Selles tabelis arvutatakse viitesuhtes ja viitesuhteta paaridele eraldi täpsus ja saagis. See tekitab terminoloogilist segadust, kuid vastavad valemid ja selgitused on toodud allpool. Samast raportist leiab ka **makro keskmise**, mis võtab kummagi paariliigile eraldi arvutatud mõõdiku keskmise. Makro keskmine ei arvesta testandmestiku tegeliku koostisega ehk ei vaata, mitu viitesuhtes ja viitesuhteta paari andmestikus on. **Kaalutud keskmine** aga arvestab arvutamisel paariliikide tasakaaluga andmestikus ja on seetõttu parem näitaja, kui testandmestikus on viitesuhteta paare kordades rohkem, kui viitesuhtes paare.

c) $täpsus = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$

d) $saagis\ ehk\ sentsitiivsus = \frac{TP}{TP+FN}$

e) $F1 = 2 * \frac{täpsus * saagis}{täpsus + saagis}$

f) $viitesuhteta\ täpsus\ = \frac{TN}{TN+FN} = tegelikult\ negatiivne\ ennustusväärtus$

g) $viitesuhtes\ täpsus = täpsus = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$

h) $viitesuhteta\ saagis = \frac{TN}{TN+FP} = tegelikult\ tõeselt\ negatiivsete\ määr\ ehk\ spetsiifilisus\ (ingl\ \mathrm{specificity})$

i) $viitesuhtes\ saagis = saagis = \frac{TP}{TP+FN}$

j) $viitesuhtes\ F1 = 2 * \frac{viitesuhtes\ täpsus * viitesuhtes\ saagis}{viitesuhtes\ täpsus + viitesuhtes\ saagis}$

k) $viitesuhteta\ F1 = 2 * \frac{viitesuhteta\ täpsus * viitesuhteta\ saagis}{viitesuhteta\ täpsus + viitesuhtes\ saagis}$

Õigsus (ingl *accuracy*) näitab, kui palju ennustatud paaridest ennustati korrektselt. Õigsust vaadatakse nii testandmestikul kui ka treening- ja valideerimisandmestikul. Viimast kaht kujutatakse graafikul, et näha selle muutust treenimise käigus.

l) $õigsus = \frac{TP+TN}{N+P}$

[8] Binaarse klassifikatsiooni korral võib seda nimetada ka sensitiivsuseks (ingl *sensitivity*).

MCC (ingl *Matthews correlation coefficient*) võtab arvesse kõik sagedusmaatriksi lahtrid ja tagastab arvu vahemikus -1 ja +1. +1 tähendab ideaalset ennustamist, 0 suvalist ennustamist ja -1 vastupidi ennustamist (iga viitesuhtes paarile ennustatakse viitesuhteta olemine).

m) $MCC = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

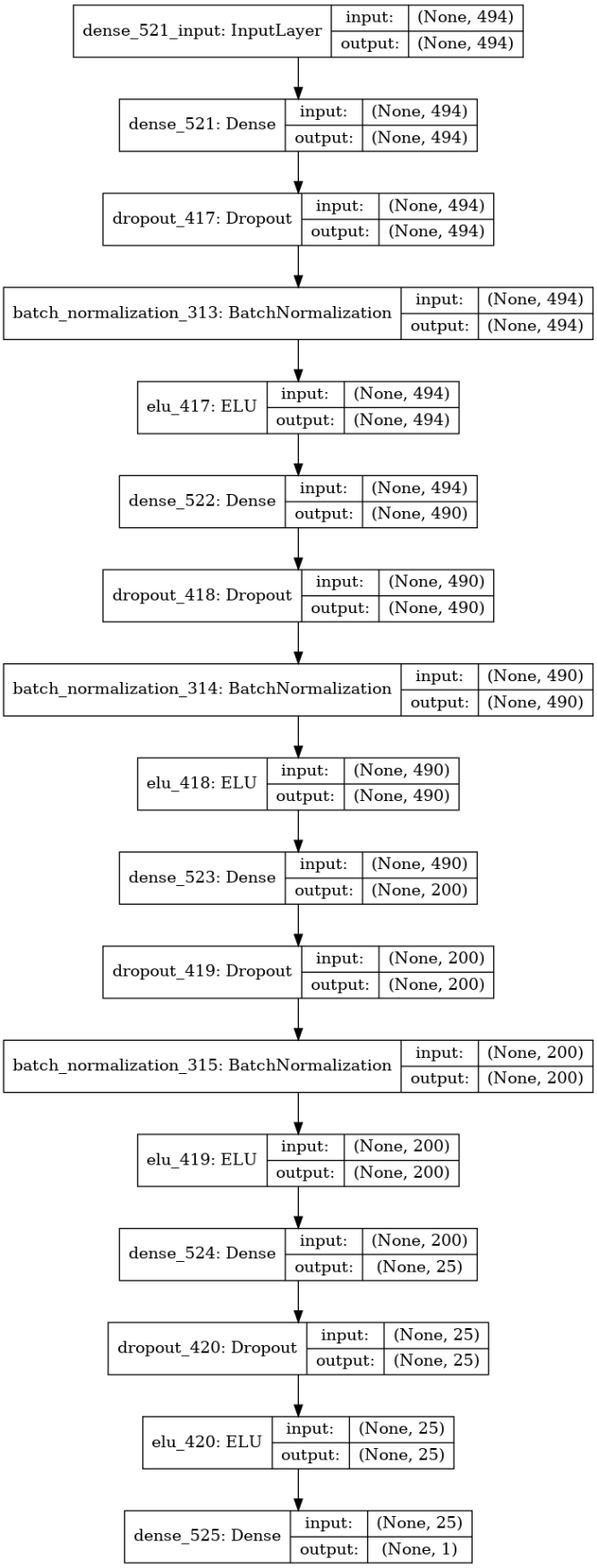
Kahju (ingl *loss*) väärtus sõltub sellest, millist kahjufunktsiooni kasutatakse. Siin töös kasutatakse binaarset ristentroopiakahjut (ingl *binary crossentropy loss*). Kahju arvutab iga testandmsestiku paari kohta, kui kaugel oli ennustatud väärtus tõest (nullist või ühest) ja võtab kõikide paaride keskmise. Sellest võib mõelda, kui mõõdikust, mis näitab, kui valed antud närvivõrgu ennustused on. Kahjut hinnatakse nii testandmestikul kui ka treening- ja valideerimisandmestikul treenimise vältel.

n) $binaarne\ ristentroopiakahju\ L(y,\hat{y}) = -\frac{1}{N}\sum_{i=0}^{N}(y * \log(\hat{y}_i) + (1-y) * (\log(1-\hat{y}_i))$, kus y on tõesed väärtused ja $\hat{y}$ ennustatud väärtused.

Kuna närvivõrgud on väga stohhastilised ehk juhusest sõltuvad, annab üks ja sama närvivõrk igal treenimisel (ka samal andmestikul) erineva tulemuse. Selleks, et hinnata närvivõrgu üldist tulemust, treenitakse üht ja sama närvivõrku mitu korda ja vaadatakse mõõdikute keskmisi tulemusi arvestades 95protsendilist usaldusintervalli. **95%-line usaldusintervall** (ingl *95 percent confidence interval*) näitab, millises vahemikus 95% tõenäosusega antud mõõdiku tegelik keskmine asub. Varieeruvuse huvides treenitakse närvivõrke x korda **k-korda ristvalideerimise meetodil** (ingl *k-fold cross-validation*) (kokku x*k närvivõrku), kus igas x_i korras valitakse uus hulk negatiivseid näiteid.

3.2. Edukamad närvivõrgud

Närvivõrkude NNa ja NN1 tulemuste hindamiseks eraldati andmestiku lõpust umbes 20% testandmestiku tarvis ning ülejäänu põhjal treeniti mõlemat närvivõrku 15 korda 7-korra ristvalideerimise meetodil. Kuna andmestik on tasakaalust väga väljas, moodustati iga uue 7-korra ristvalideerimise jaoks uus andmestik juhuslikult valitud 51010 negatiivsest näitest ehk viitesuhteta paarist (0) ja 5095 positiivsest näitest ehk viitesuhtes paarist (1), et proovida närvivõrku treenida võimalikult paljudel erinevatel



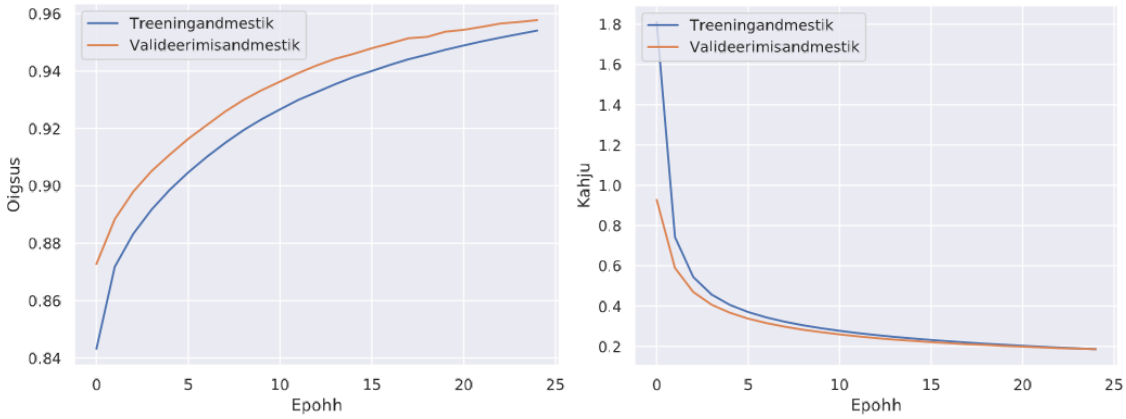
Joonis 1. **NN1 ja NNa ülesehitus.**

negatiivsetel näidetel. Positiivseid näiteid ei valitud juhuslikult, kuna neid oligi koguandmestikus kokku 6230, millest 1135 jäid testandmestiku jaoks. Iga positiivset näidet lisati andmestikku 10 korda, mistõttu oli sisendiks kategooriate suhe peaaegu 1:1. Selle otsusega seoses tuleb meeles pidada, et positiivsete näidete kordamisel võib sattuda treening- ja valideerimisandmestikku üks ja sama positiivne näide. Iga treeningu pikkus oli 25 epohhi ja ploki suurus 256.

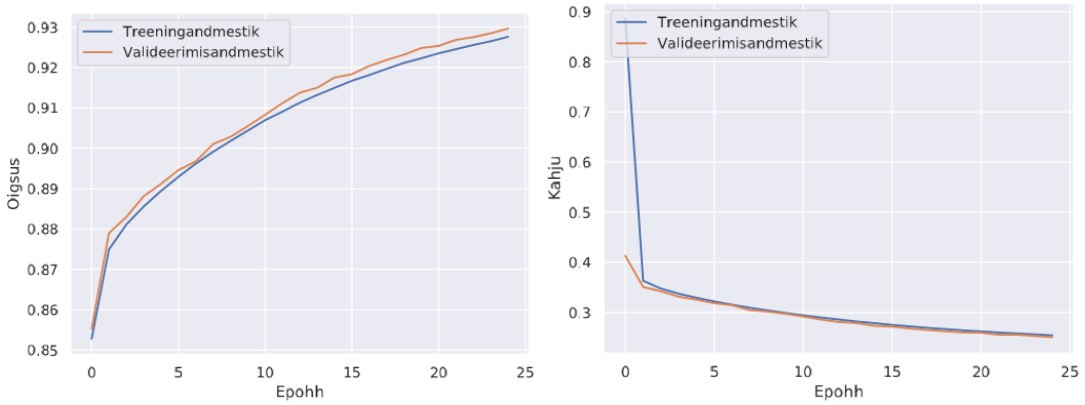
Joonistel 2 ja 3 on näha NNa ja NN1-e keskmiseid õigsuse ja kahju muutuseid treening- ja valideerimisandmestikul treenimise vältel. Nendest on näha, et NNa jõuab NN1-st vähemate epohhidega madalamale kahjule ja kõrgemale õigsusele. Samuti on NNa lõpptulemused paremad. NN1 treening- ja valideerimisandmestiku tulemused ei eristu mõlema mõõdiku puhul teineteisest väga – see tähendab, et mudel ei muutu treeningandmestiku spetsiifiliseks

(vastasel juhul oskab mudel treeningandmestiku peal hästi ennustada, aga puudub uute näidete ennustamiseks üldistusvõime). NNa keskmise õigsuse puhul on näha erinevust treening- ja valideerimisandmestiku tulemustes. See siiski väheneb epohhide lõpus ega ole liiga suur.

Kahju jääb treening- ja valideerimisandmestikul mõlemal mudelil alla 0,3-e (NNa-l on kahju testandmestikul 0,26 ja NN1-l 0,29). Seda võib pidada madalaks, kui arvestada eelnevate katsete kõrgemaid kahjusid ja asjaolu, et väga madalat kahjut ei saagi oodata – tegu on keerulise ülesandega, kus käsitsi märgendadeski tekkis küsimusi ja arutelusid. Siiski võiks see ideaalis olla madalam. Mõlema mudeli kahjud treening- ja valideerimisandmestikul pea ühel joonel. See viitab sellele, et puudub ülesobitamise (ingl *overfitting*) probleem.



Joonis 2. **NNa õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**



Joonis 3. **NN1 õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

Tabelist 2 ja 3 näeb mõlema närvivõrgu treenimisandmeid ja tulemusi. Seal on näha, et NNa ja NN1 tulemused on üsna sarnased, kuid NNa edestab keskmise MCC mõõdikus NN1-e 0,03 punkti võrra ning keskmises õigsuses ja kaalutud F1-s 0,01 võrra. Teisteski mõõdikutes saab NNa NN1-ga võrdseid või sellest paremaid tulemusi. Mõlema närvivõrgu kaalutud keskmine F1 on siiski silmapaistev (0,93–0,94). Ka õigsus ületab ootusi jäädes 0,9 ja 0,92 vahele. Seega võib öelda, et NNa annab NN1-st veidi paremaid tulemusi ja järelikult on optimeerimisalgoritmina Adagrad edukam. Märkimisväärselt parem on see vaid MCC mõõdiku ja madalama kahju suhtes.

Tabel 2. **NNa treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NNa**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	**0,92** 0,921–0,923	0,26 0,26–0,266	**0,47** 0,466–0,471
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	0,99 0,99–0,99	0,92 0,922–0.926	0,96 0,958–0,96
Treeniti x korda k-korda	x = 15 k = 7	Viitesuhtes	0,29 0,287–0,293	0,85 0,847–0,852	0,43 0,429–0,436
Ploki suurus	256	Makro keskmine	0,64 0,641–0,644	0,89 0,886–0,888	0,7 0,693–0,697
Epohhe	25	Kaalutud keskmine	0,97 0,97–0,97	0,92 0,921–0,923	**0,94** 0,939–0,941

Tabel 3. **NN1 treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NN1**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	**0,91** 0,904–0,91	0,29 0,284–0,297	**0,44** 0,433–0,442
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	0,99 0,991–0,993	0,91 0,906–0,912	0,95 0,948–0,952
Treeniti x korda k-korda	x = 15 k = 7	Viitesuhtes	0,26 0,252–0,264	0,85 0,848–0,859	0,39 0,387–0,4
Ploki suurus	256	Makro keskmine	0,63 0,623–0,629	0,88 0,878–0,882	0,67 0,668–0,676
Epohhe	25	Kaalutud keskmine	0,97 0,97–0,97	0,91 0,904–0,91	**0,93** 0,928–0,932

Kaalutud keskmine F1, täpsus, saagis ning keskmine õigsus on mõlema närvivõrgu puhul ideaalilähedaselt üle 0,9, muret tekitab ainult I tüübi viga. NN1 I tüübi viga on 0,75, II tüübi viga 0,01 (vt tabelit 4). NNa I tüübi viga on 0,71, II tüübi viga 0,01 (vt tabelit 5). See tähendab, et mõlema närvivõrgu ennustustest on viitesuhte ennustuse saanud paaridest on rohkem valepositiivseid, kui tõeselt positiivseid. See viga on üsna kõrge ja murettekitav – keeletehnoloogia tööstuses ei saa kasutusele võtta nii suure veaga mudelit.

Tabel 4. **NNa keskmine segadusmaatriks.**

I tüübi viga	**0,71**	**Mudeli ennustus**	
II tüübi viga	0,01	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	964,18 85%	170,82 15%
	Viitesuhteta 100%	2369,1 8%	28982,9 92%

Tabel 5. **NN1 keskmine segadusmaatriks.**

I tüübi viga	**0,75**	**Mudeli ennustus**	
II tüübi viga	0,01	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	968,15 85%	116,85 15%
	Viitesuhteta 100%	2842,81 9%	28509,19 91%

Kõrges I tüübi veas võib rolli mängida ka positiivsete ja negatiivsete näidete osakaal juhuslikult valitud (andmestiku lõpust 20%) testandmestikus – nende suhe on 1:26. Negatiivsete näidete suur hulk ja positiivsete näidete madal hulk tõstab võimalust I tüübi vea tõusuks ning viitesuhtes F1-e ja täpsuse madalamaks tulemuseks. Testides neid närvivõrke tasakaalu huvides testandmestikuga (vt lisa 1), kus on võrdselt positiivseid ja negatiivseid näiteid (neid on siiski vähe: 1135 positiivset ja sama palju

negatiivset näidet), tõusevad keskmised viitesuhtes täpsused ja F1-d nendel närvivõrkudel 0,45–0,65 võrra. Samuti teeb märkimisväärse hüppe MCC, mis NN1-l tõuseb 0,77-ni ja NNa-l 0,78-ni. Segadusmaatriks jäi üsna sarnaseks, kuid loogiliselt (nüüd on valepositiivsete arv väiksem, mis siis, et protsentuaalselt sama) on I tüübi viga väiksem mõlema närvivõrgu puhul (NNa-l 0,07, NN1-l 0,09). Samas tõusis mõlema närvivõrgu II tüübi viga 0,15-ni.

Mainimata jäänud mõõdikud halvenesid. Kahju tõusis mõlema närvivõrgu puhul 0,42-ni. Keskmine õigsus, kaalutud keskmine täpsus, saagis, F1 ja viitesuhteta täpsus ja F1 langevad veidi (v.a kaalutud keskmine täpsus, mis langeb palju). Samas jäävad need mõõdikud siiski üsna kõrgele tasemele, näiteks NN1 õigsus on 0,88 ja NNa õigsus 0,89. Ka tasakaalus testandmestikul edestab NNa (Adagradiga) napilt NN1-e (Adamiga).

Erinevused erinevatel testandmestikel illustreerivad närvivõrkude hindamise keerulisust. Siiski võiks esimest, tasakaaluta varianti pidada tõetruumaks. Võttes ette uue teksti, millest leitakse nende võrkude abil üles pronominaalsete viitesuhete paarid, kasutades peatükis 2.3 kirjeldatud paaride eraldamistingimusi, tekib loogiliselt võttes alati rohkem negatiivseid kui positiivseid näiteid ja seetõttu on tasakaaluta testandmestikul saadud tulemused realistlikumad. Ka ülejäänud siin peatükis saadud tulemused on sel põhjusel arvutatud tasakaaluta testandmestiku peal.

NN1 ja NNa-d saab kasutusele võtta treenides neid mitu korda ja ennustada tulem mudelite ansambliga (ingl *ensemble of models*) või treenides juhuslikult ühe ja kasutada seda. Github'is salvestatud mudeleid ei ole.

3.3. Tunnuste kodeerimise mõju

Tähelepanelik lugeja mõtles peatüki 2.3 tunnuste eraldamise kohta lugedes kindlasti esimese asjana tunnuste kodeerimise peale. Närvivõrkude treenimisel on oluline tunda oma andmestikku ning normaliseerida tunnuseid nii, et tunnuse keskmine jääks nulli lähedale. Kui tunnused on erinevatel skaaladel (näiteks kaugus sõnedes vahemikus [-300, 300] ja kaugus lausetes vahemikus [-3,3]), võib juhtuda, et mõned kaalud närvivõrgus uuenevad kiiremini, kui teised, ja närvivõrgu õpiprotsess on aeglasem. Samuti võivad suurema skaalaga tunnused omada suuremat mõju väljundile. (Vt nt

(Stöttner 2019)) Seetõttu on oluline mõelda sellele, kuidas tunnuseid vektoris esitatakse.

Antud töö andmestikus on kolme sorti tunnuseid: binaarsed (jah-ei), kategoriaalsed (kindel arv järjekorrata kategooriaid nagu sõnaliik ja kääne) ja arvandmed (kaugus sõnedes, lausetes). NN1 ja NNa kasutasid kõigi mitte-binaarsete tunnuste kodeerimisel *1-hot-encoding*-ut, mis sobib rohkem kategoriaalsete tunnuste teisendamiseks. ***1-hot-encoding***[9] võtab kategoriaalse tunnuse nagu viitealuse asukoht, millel on kolm kategooriat (*lause alguses, lause keskel* ja *lause lõpus*) ja teisendab need kolmeks uueks binaarseks tunnuseks (*on_lause_alguses, on_lause_lõpus, on_lause_keskel*). See pole kõige mõistlikum lähenemine, kui kategooriatel on järjestus oluline või kui tunnused on hoopis arvandmed (*kaugus_asendussõnast_on_1, kaugus_asendussõnast_on_2…*).

Arvandmete normaliseerimiseks kasutatakse näiteks **min-max normaliseerimist**[8] (ingl *min-max normalization, min-max scaling*), mis teisendab tunnuse skaalale [0,1].

o) $min - max\ normaliseerimine = \frac{X_i - \min(X)}{\max(X) - \min(X)}$, kus min(X) on tunnuse minimaalne väärtus, max(X) tunnuse maksimaalne väärtus ning X_i see väärtus, mida normaliseeritakse.

Erinevate kodeeringute võrdlemiseks koostati kolm katset, mille keskmisi tulemusi mõõdeti tasakaaluta testandmestiku peal. **NNa_minmax** on NNa-ga sarnane. Ainuke erinevus on esimeses kihis: kuna arvandmed on normaliseeritud min-max'iga, on tunnuste vektoris 494 tunnuse asemel 268 tunnust. **NN_minmax** on natukene ka varjatud katse uurida närvivõrgu laiuse mõju. See on mugavdatud versioon NNa_minmax-ist, kus kihid pole enam nii laiad (kuna tunnuste arv vähenes) – vastavalt 268-268-264-110-20 sõlme (ingl *node*) (NNa_minmax'i 268-490-200-25 asemel). Kolmas katse **NN_nocoding** testib närvivõrku, mille tunnuseid polegi kodeeritud. See on teistest kõige kitsam (vastavalt 32-40-38-30-8 sõlme) ning saab sisendiks algsed 32 tunnust muutmata kujul. Lisades 2–4 on toodud nende närvivõrkude treenimisandmed ja tulemused.

Tulemustest on näha, et veidi kitsam närvivõrk annab sama kodeeringuga sisendiga

[9] Vt nt scikit-learn'i kasutusjuhendis andmete eeltöötlemise peatükki: https://scikit-learn.org/stable/modules/preprocessing.html

natukene madalamaid tulemusi. Näiteks NNa_minmax'i keskmine MCC on 0,43, keskmine õigsus 0,9 ja kaalutud keskmine F1 0,92. NN_minmax'i MCC on 0,41, õigsus 0,88 ja kaalutud keskmine F1 0,92. NN_minmax'i I tüübi viga ja kahju on 0,02 võrra kõrgem. Segadusmaatriksitest võib näha, et kitsam NN_minmax ennustab viitesuhtes paare NNa_minmax'ist natukene paremini ja viitesuhteta paare halvemini.

Kõige kitsam NN_nocoding saab kahe eelneva närvivõrguga võrreldes palju halvemaid tulemusi. Selle MCC on 0,26, õigsus 0,76 ja kaalutud keskmine F1 0,84. Kas I tüübi viga on väga suur – 89% ennustatult viitesuhtes paarides on tegelikult viitesuhteta. Vaid 76% viitesuhteta paaridest saab korrektse ennustuse. See närvivõrk ei suuda võistelda NN1 ja NNa-ga. Madalad tulemused sõltuvad kindlasti nii tunnuste kodeerimatusest kui ka närvivõrgu kitsusest.

NNa_minmax'i ja NN_minmax'i tulemused näitavad, et laiema närvivõrgu tulemused on paremad ka siis, kui sisendvektor on lühem. NNa_minmax'i õigsus ja kaalutud keskmine F1 jäävad NNa-le alla 0,02 punkti võrra, kahju ja MCC 0,04 võrra ja I tüübi viga 0,05 võrra. Ainuke mõõdik, kus NNa_minmax NNa-st parem on, on viitesuhtes paaride ennustused: 88% (85% asemel) said õige ennustuse. Samas viitesuhteta paaridest ennustati õigesti 90% (92% asemel). Seega võib öelda, et arvandmetel min-max normaliseerimise kasutamine *1-hot-encoding*'u asemel annab nende testide põhjal veidi halvemaid tulemusi.

Nende katsete edasiarendusena tasub proovida ka teisi kodeerimisviise. Näiteks min-maxi asemel kasutada arvandmete vahemike kategooriate loomist (*kaugus_asendussõnast_on_kuni_5, kaugus_asendussõnast_on_5_kuni_10, kaugus_asendussõnast_on_üle_150ne).* Magistritöö raames rahuldutakse aga *1-hot-encoding*'uga saadud tulemustega.

3.4. Treening- ja valideerimisandmestiku viitesuhtes ja viitesuhteta paaride osakaalu ja suuruse mõju

Kui korpusest saadud andmestiku lõpust eraldada ca 20% testandmestiku jaoks, jääb treening- ja valideerimisandmestikule 123846 viitesuhteta paari ja 5095 viitesuhtega paari. See teeb 24 negatiivset näidet ühe positiivse näite kohta. Tegu on väga tasakaalust

väljas andmestikuga. Eelnevad närvivõrgud on närvivõrke treenitud andmestikul, kus iga positiivne näide on andmestikku lisatud 10 korda ning sama palju on lisatud erinevaid juhuslikult negatiivseid näiteid (suhe on 1:1). See alapeatükk uurib, kuidas muutub närvivõrgu tulemus, kui treening- ja valideerimisandmestiku kogust ja paaride suhet muuta.

Olemasoleva andmestikuga mängides korraldati neli katset NNa põhjal. **NNa_alldata** on NNa närvivõrk, mis saab sisendiks kogu olemasoleva andmestiku (5095:123846), millest juhuslikult valitud kaheteistkümnendik (10745 paari) eraldatakse valideerimisandmestiku tarbeks. **NNa_smallequaldata** saab sisendiks ühekordse portsu positiivseid näiteid ja sama palju juhuslikult valitud negatiivseid näiteid (5095:5095), millest juhuslikult valitud kümnendik (1091) läheb valideerimisandmestikku. Viitesuhteta näidete suurema osakaalu mõju uurib **NNa_1pos2neg**, mis saab samuti sisendiks ühekordse portsu positiivseid näiteid, aga kolmekordse portsu negatiivseid juhuslikult valitud näiteid (5095:15285, sellest kaheksandik (2547) läheb valideerimisandmestikku). Neljas katse **NNa_5equaldata** uurib, kas suhtega 1:1 suuremal andmestikul (aga NNa-st väiksemal andmestikul) närvivõrgud saavad paremaid tulemusi. NNa_5equaldata sisendiks on 5 korda 5095 positiivset näidet ning sama palju juhuslikult valitud negatiivseid näiteid (5*5095:25475, millest seitsmendik (7278) läheb valideerimisandmestikku).

Kogu andmestikul treenitud NNa(_alldata) keskmised tulemused ületavad märgatavalt kümnekordselt positiivsetel ja sama palju negatiivsetel treenitud NNa tulemusi: MCC tõuseb 0,47-lt 0,59-le, keskmine õigsus ja kaalutud keskmine F1 on ideaalilähedaselt 0,98 ja 0,97, kahju langeb 0,08-ni (vt lisa 5). Segadusmaatriksit uurides on aga aru saada, et närvivõrk pole kaugeltki nii ideaalne, kui tundub. Kõrged skoorid erinevates mõõdikutes tulenevad asjaolust, et närvivõrk õppis hästi viitesuhteta paare tundma – 99% viitesuhteta paaridest ennustati õigesti. Viitesuhtes paaride osas on see närvivõrk väga segaduses, tuvastades vaid pooled paaridest õigesti. Tulemusena ennustab närvivõrk enamasti paaridele viitesuhteta olekut ja saab testandmestiku tasakaalutuse tõttu kõrgeid skoore, kuid tuvastab vaid 50% viitesuhtes paaridest õigesti ning 27% viitesuhte ennustuse saanud paaridest on tegelikult viitesuhteta (vt tabelit 6). Ka lisas 5 toodud õigsuse muutused treening- ja valideerimisandmestikul epohhide lõikes näitavad

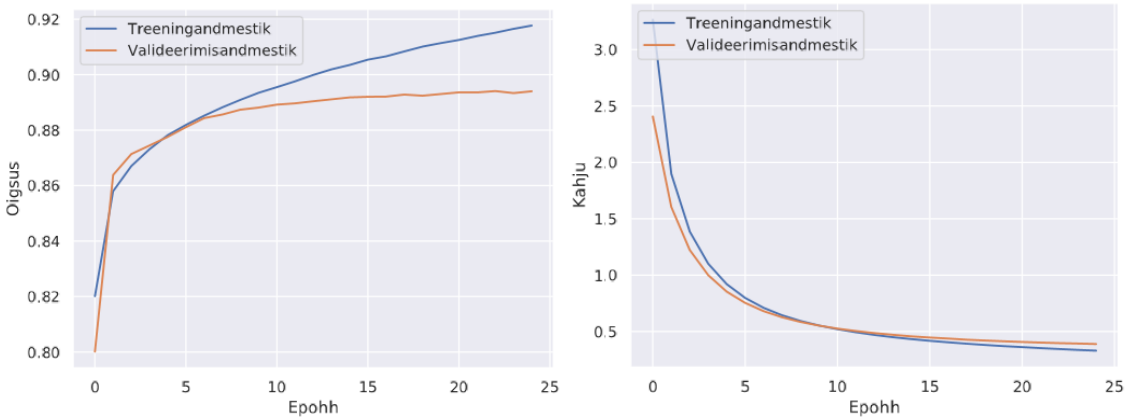
tulemuste muutumatust valideerimisandmestikul, samas kui treeningandmestiku tulemused näiliselt tõusevad. Seetõttu ei saa NNa_alldata't pidada NNa-st paremaks.

Veidi vähem tasakaalust väljas oleval andmestikul (1:3 suhtes, aga see-eest palju väiksemal andmestikul) treenitud närvivõrk NNa_1pos3neg tundub esmapilgul NNa-ga sama hea olema: keskmine õigsus ja MCC on 0,01 võrra suuremad, F1 sama, kuid kõrgema usaldusvahemikuga. I tüübi viga on samuti väiksem. Vaid kahju on 0,04 võrra suurem (vt tulemusi lisas 6). Samas on jooniselt 4 näha õigsuse kiiremat tõusu treeningandmestikul ning aeglasemat tõusu (pigem seisma jäämist) valideerimisandmestikul epohhide lõikes. See näitab närvivõrgu üldistusvõime kadu ja liigset treeningandmestikule spetsialiseerumist. Ka segadusmaatriksist (tabel 7) on näha, et viitesuhtes paaride ennustamise edukus vähenes 4% võrra. Seega võib öelda, et nõrgalt tasakaalust väljas oleva märgatavalt väiksema andmestiku peal NNa treenimisel jäävad tulemused näiliselt samaks, kuid väheneb viitesuhtes paaride tuvastamisoskus, tõuseb kahju ja treenimisel ei käi õigsus treening- ja valideerimisandmestikul ühte sammu. Siin võib rolli mängida ka suur epohhide arv väiksel andmestikul – viiendal epohhil on treening- ja valideerimisandmestiku õigsus võrdne, ainult kahju on kõrge (ca 0,75). Väiksema andmestikuga töötades tasub proovida vähemate epohhidega treenimist või varast lõpetamist (ingl *early stopping*, vt nt (Prechelt 2012)).

Kui võrrelda NNa_smallequaldata't (vt lisa 7) NNa_1pos3neg'iga (vt lisa 6), on näha viimase paremust keskmises õigsuses (0,93 vs 0,88), MCC-s (0,48 vs 0,4) ja kaalutud keskmises F1-s (0,94 vs 0,91). Kui NNa_smallequaldata ennustab nii viitesuhtes kui ka viitesuhteta paaridest 88% õigesti, siis NNa_1pos3neg'i puhul on erinevus suurem (vt tabel 7). Kõrgem viitesuhteta paaride saagis (93%) seletab mõõdikute kõrgemaid tulemusi hoolimata sellest, et viitesuhtes paaride saagis on NNa_smallequaldata'st 7% võrra madalam (81%).

Tabel 6. **NNa_alldata keskmine segadusmaatriks.**

I tüübi viga	0,27	**Mudeli ennustus**	
II tüübi viga	0,02	Viitesuhtes	Viitesuhteta
Tege-likult	Viitesuhtes 100%	572,92 **50%**	562,08 **50%**
	Viitesuhteta 100%	216,18 1%	31135,82 99%

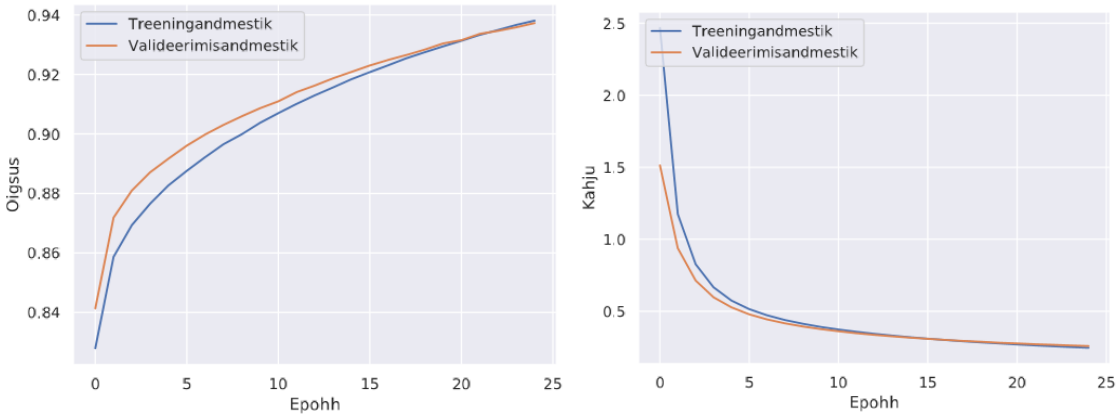


Joonis 4. **NNa_1pos3neg õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

Tabel 7. **NNa_1pos3neg keskmine segadusmaatriks.**

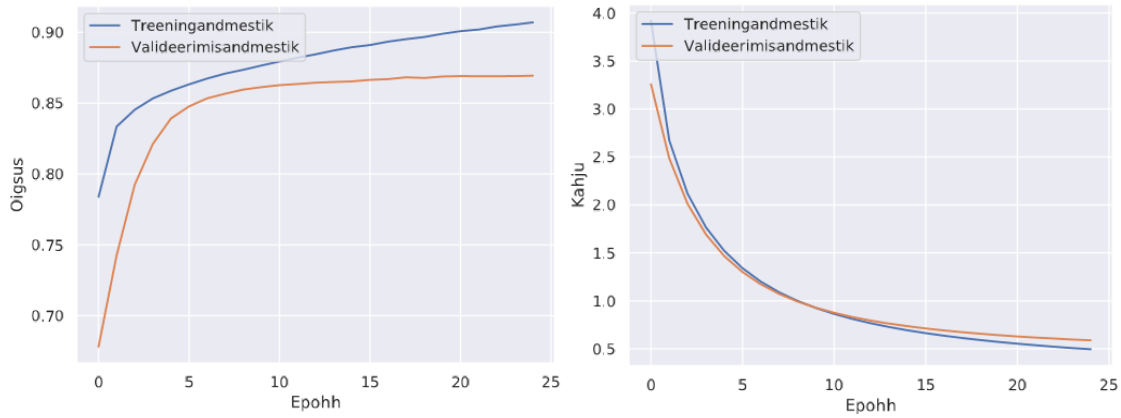
I tüübi viga	0,69	**Mudeli ennustus**	
II tüübi viga	0,01	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	924,08 81%	210,92 19%
	Viitesuhteta 100%	2070,08 7%	29281,92 93%

Tasakaalus sisendi suuruse mõju näitavad NNa_smallequaldata ja NNa_5equaldata (vaata nende tulemusi tasakaaluta testandmestikul lisades 7 ja 8). Neid omavahel võrreldes on näha NNa_5equaldata kerget paremust, mis tuleneb 2% võrra suuremast viitesuhteta paaride ära tundmisest. Ka õigsuse muutuse järgi treening- ja valideerimisandmestikul võib selgelt eelistada NNa_5equaldata't, kuna viimasel käivad õigsused ühte sammu ja on lõpuepohhides võrdsed (vt joonist 5). NNa_smallequaldata puhul on märgata kahju aeglasemat madaldumist ning valideerimisandmestikul mõõdetud õigsuse kasvu seismajäämist ja erinevust treeningandmestikust (vt joonist 6).



Joonis 5. **NNa_5equaldata õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

NNa'ga (mille alternatiivne nimi võiks olla NNa_10equaldata) võrreldes ei jõua NNa_5equaldata õigsus treenimisel kõrgele, kuid on treening- ja valideerimisandmestikul meeldivalt sarnasem. Huvitav on märkida, et kuigi NNa keskmised tulemused kahju (0,26 vs 0,33), õigsuse (0,92 vs 0,9), MCC (0,47 vs 0,43) ja F1 (0,94 vs 0,93) vallas on veidi paremad, tuvastab NNa viitesuhtes paare NNa_5equaldata'st halvemini (aga viitesuhteta paare paremini – st „käärid" on suuremad). Tabelist 8 on näha, et kuigi NNa_5equaldata I tüübi viga (0,76) on NNa I tüübi veast (0,71) suurem, tuvastab see 88% viitesuhtes paaridest ja 90% viitesuhteta paaridest õigesti (s.o NNa-st rohkem viitesuhtes paare ja vähem viitesuhteta paare, NNa edukus mõõdikutes tuleneb kõrgemas viitesuhtes paaride tuvastamises).



Joonis 6. **NNa_smallequaldata õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

Tabel 8. **NNa_5equaldata keskmine segadusmaatriks.**

I tüübi viga	0,76	**Mudeli ennustus**	
II tüübi viga	0,0	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	995,66 88%	139,34 12%
	Viitesuhteta 100%	3079,83 10%	28272,17 90%

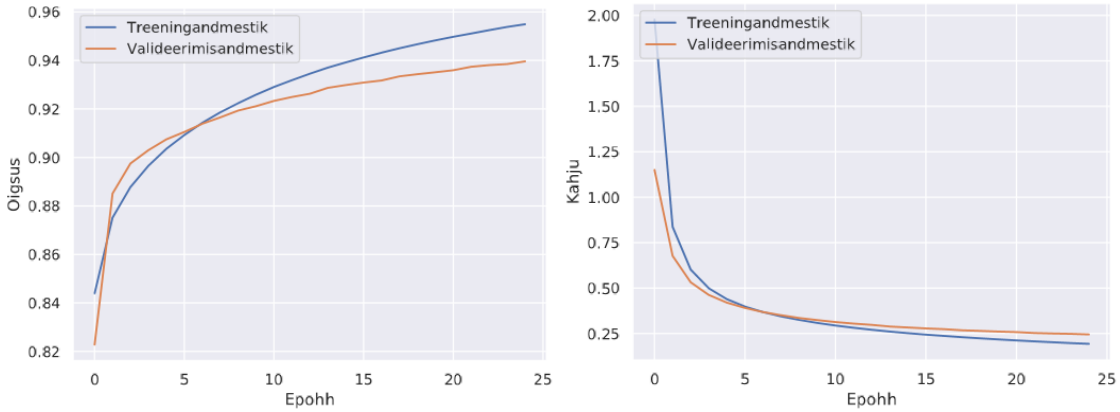
Siin kerkib küsimus, millist närvivõrku eelistada. Kui soovida viitesuhtes paaride kõrgemat tuvastust, aga leppida ennustustes suurema hulga valepositiivsete paaridega, võib eelistada NNa_5equaldata't. Viimase kasuks räägib ka sarnasem õigsuse areng epohhide lõikes treening- ja valideerimisandmestikul. Kui eelistada väiksemat valepositiivide arvu, aga leppida suurema valenegatiivide arvuga (st. mudel saab vähem viitesuhteid kätte, aga viitesuhte ennustuse saanud paaride seas on vähem praaki), sobib NNa paremini. Seega võib öelda, et 1:1 suhtes andmestiku suurendamine positiivsete kordamise näol tõstab tulemust pigem viitesuhteta paaride äratundmise näol (neid ju ei korratud) ning vähendab viitesuhtes paaride tundmise täpsust – mudel ennustab rohkem viitesuhteta paare – ja NNa_5equaldata't võib pidada samuti üheks edukamaks närvivõrguks.

Kui andmeid on liiga vähe, tasub proovida andmete rikastamise (ingl *data augmentation*) (vt nt Goodfellow jt 2016: 236–238) meetodeid. Andmete rikastamine tähendab andmete juurde genereerimist olemasolevate andmete vähesel muutmisel (näiteks pildi ümber keeramist, sellele müra lisamist). Siin katsetatakse positiivsete näidete juurde genereerimist ADASYN-i abil. **ADASYN** (Haibo He jt 2008) võtab iga vähemuses oleva kategooria näite (viitesuhtes paari), leiab andmestikust sellele 5 kõige lähedasemat näidet ja genereerib iga lähedasema näite ja kategooria näite vahele uue pseudonäite vähemuses kategooriasse vastavalt sellele, kui palju uusi näiteid juurde soovitakse.

ADASYN-i testimiseks loodi **NNa_adasyn**, mis treenib NNa_5equaldata'ga sarnaselt 1:1 suhtega andmestikul, kus positiivseid näiteid on lisatud viis korda. Erinevusena on positiivsete paaride arvu suurendatud ADASYN-i abil 3385 võrra: nii on positiivsete

treening- ja valideerimisandmestiku näidete hulk 7% negatiivsetest treening- ja valideerimisandmestiku näidetest (varem oli ca 4%). Uus treeningandmestik on suhtes 5*8480:42400, millest seitsmendik (12114) läheb valideerimisandmestikku. Vaata selle tulemusi lisas 9.

Jooniselt 7 on näha NNa_adasyn'i õigsuse ja kahju muutuseid treening- ja valideerimisandmestikul epohhide lõikes. NNa tulemustega (vt joonis 2) võrreldes on kohe näha, et NNa_adasyn'i kahju jääb NNa-st kõrgemale (samuti on näha väikest erinevust kahjudes erinevate andmestike lõikes) ning õigsus ei käi treening- ja valideerimisandmestikul ühte sammu: valideerimisandmestiku tulemused jäävad testandmestiku tulemustest maha ja „käärid" laienevad (mitte ei kitsene nagu NNa puhul).



Joonis 7. **NNa_adasyn õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

NNa_adasyn'i keskmine õigsus (0,89), MCC (0,45), kaalutud F1 (0,92) ja I tüübi viga (0,76) jäävad oma kaksiku, NNa_5equaldata tulemustest 0,01 võrra alla. Segadusmaatriks (tabel 9) näitab, et NNa_adasyn määrab õigesti nii 89% viitesuhtes paaridest kui ka 89% viitesuhteta paaridest. See tulemus on üsna sarnane NNa_5equaldata'ga, mis määras 88% viitesuhtes paaridest ja 90% viitesuhteta paaridest õigesti. Seega võib öelda, et ADASYN'i kasutamine tõstis 1% võrra viitesuhtes paaride tuvastamist ja langetas 1% võrra viitesuhteta paaride tuvastamist. NNa_5equaldata kerge paremus teiste mõõdikute põhjal on tingitud testandmestiku viitesuhtes ja viitesuhteta paaride arvust: 1% viitesuhteta paaridest testandmestikul (314) on suurem, kui 1% viitesuhtes paaridest (11). ADASYN'i kasutamine ei andnud mingeid

märkimisväärseid eeliseid teiste närvivõrkudega võrreldes ja selle kasutamine antud töö kontekstis on pigem ebavajalik.

Tabel 9. **NNa_adasyn keskmine segadusmaatriks.**

I tüübi viga	0,77	**Mudeli ennustus**	
II tüübi viga	0,0	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	1010,9 89%	124,1 11%
	Viitesuhteta 100%	3439,77 10%	27912,23 89%

3.5. Ploki suuruse mõju

Üks **epohh** (ingl *epoch*) tähendab, et närvivõrk vaatab ühe korra kogu treeningandmestiku läbi. Ülaltoodud närvivõrgud on seni treenitud 25 epohhiga. See tähendab, et närvivõrk vaatab treeningandmestiku 25 korda läbi. **Ploki suurus** (ingl *batch size*) on epohhi-sisene parameeter, mis määrab, mitut näidet närvivõrk vaatab enne, kui ta värskendab oma võrgu-siseseid parameetreid (kaale). Näiteks eelnevates näidetes seni kasutatud ploki suurus 256 tähendab, et ühes epohhis vaatab närvivõrk 256 esimest näidet treeningandmestikust, uuendab oma võrgu-siseseid parameetreid, vaatab järgmist 256-t näidet, uuendab uuesti, kuni treeningandmestik otsa saab. Seejärel saab alustada uut epohhi.

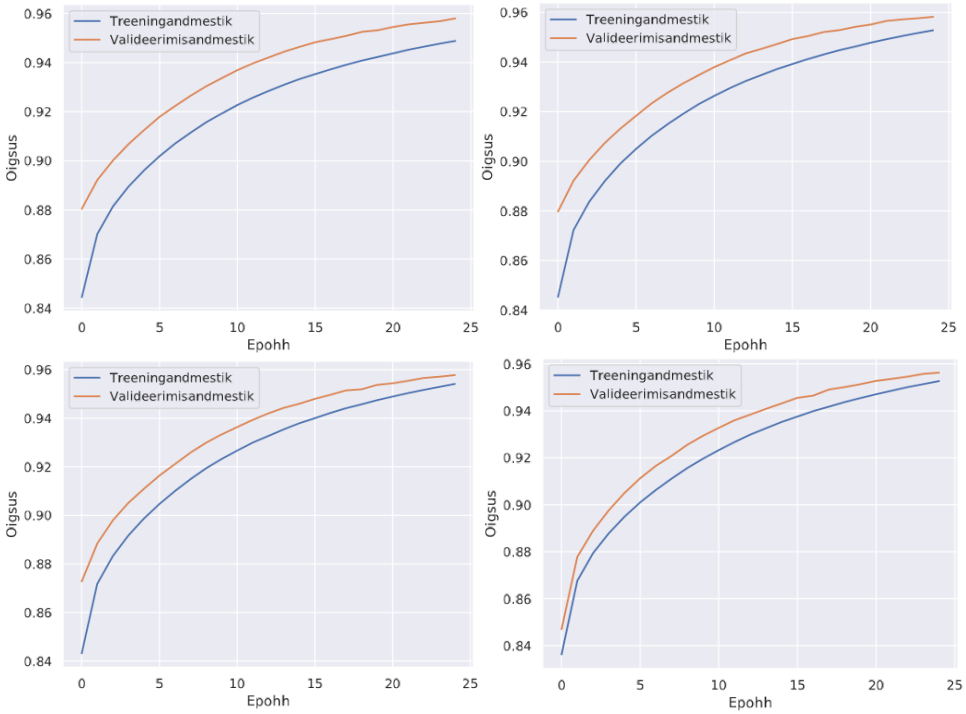
Andrew Ng sõnutsi on tüüpilisimad ploki suurused 64, 128, 256 ja 512. Samas võib ploki suuruseks võtta ühe (närvivõrk uuendab end iga näite järelt, soovitatav väga väikse, alla 2000 näitega andmestiku peal) või terve treeningandmestiku korraga (närvivõrk uuendab end igas epohhis ühe korra, treenimine on aeglasem). Praktikas kasutatakse siiski miniplokke (ingl *mini batch*), mis jäävad ühe ja kogu treeningandmestiku vahele. (Ng coursera kursus)

Ploki suuruse mõju hindamiseks tuuakse siin välja kolme katse tulemused NNa põhjal. **NNa_batch64** on NNa, mille ploki suurus on 64. **NNa_batch128** on NNa, mille ploki suurus on 128. **NNa_batch512** on NNa, mille ploki suurus on 512. Nende tulemused on

toodud lisades 10–12. Kõigi keskmised tulemused ja segadusmaatriksid on identsed NNa-ga: keskmine õigsus on 0,92, MCC 0,47, kaalutud keskmine F1 0,94. Vaid NNa_batch512-l on kahju 0,01 võrra teistest suurem (0,27). Tabelist 10 on näha, et ka segadusmaatriks on kõigil kolmel samasugune (ja võrdne NNa-ga). Seega ei mõjuta ploki suuruse muutmine NNa tulemust. Joonisel 8 võib vaadelda õigsuse muutuseid treening- ja valideerimisandmestikul erinevate ploki suurustega NNa-del. Seal on näha, et ploki suurendes väheneb õigsuse erinevus erinevatel andmestikel. Siiski ei ole mingisugust alust muuta ploki suurust, kuna tulemusi see nende katsete põhjal ei mõjutanud.

Tabel 10. **NNa_batch64, NNa_batch128 ja NNa_batch512 keskmine segadus-maatriks.**

I tüübi viga	0,71–0,72	**Mudeli ennustus**	
II tüübi viga	0,01	Viitesuhtes	Viitesuhteta
Tege-likult	Viitesuhtes	85%	15%
	Viitesuhteta	8%	92%



Joonis 8. **Erinevate ploki suurustega NNa õigsuse muutused treening- ja valideerimisandmestikul epohhide lõikes. Üleval vasakult paremale: NNa_batch64, NNa_batch128. All vasakult paremale: NNa (ploki suurus 256), NNa_batch512.**

3.6. Õpisammu mõju

Kõige olulisemaks parameetriks, millele väärtust otsida närvivõrgu parameetreid seadistades, peetakse õpisammu (Ng coursera kursus). Õpisamm on optimeerimis-algoritmi muudetav parameeter. Kui optimeerimisalgoritmi ülesanne on leida, milliste närvivõrgu parameetriga saavutatakse madalaim kahju, siis õpisamm on selle leidmise kiirus. Liiga suure sammu puhul on oht, et optimeerimisalgoritm astub madalaimast kahjust suure sammuga üle ega leiagi seda üles. Väikse sammu miinus on liiga pikk otsimisaeg või võimalus, et väike samm leiab mõne teise kahju, mida peab madalaimaks, aga mis seda tegelikult edasi vaadates ei ole. (Vt nt (Goodfellow jt 2016: 80–84, 424)) Selles alapeatükis analüüsitakse erineva õpisammuga NN1 ja NNa tulemusi.

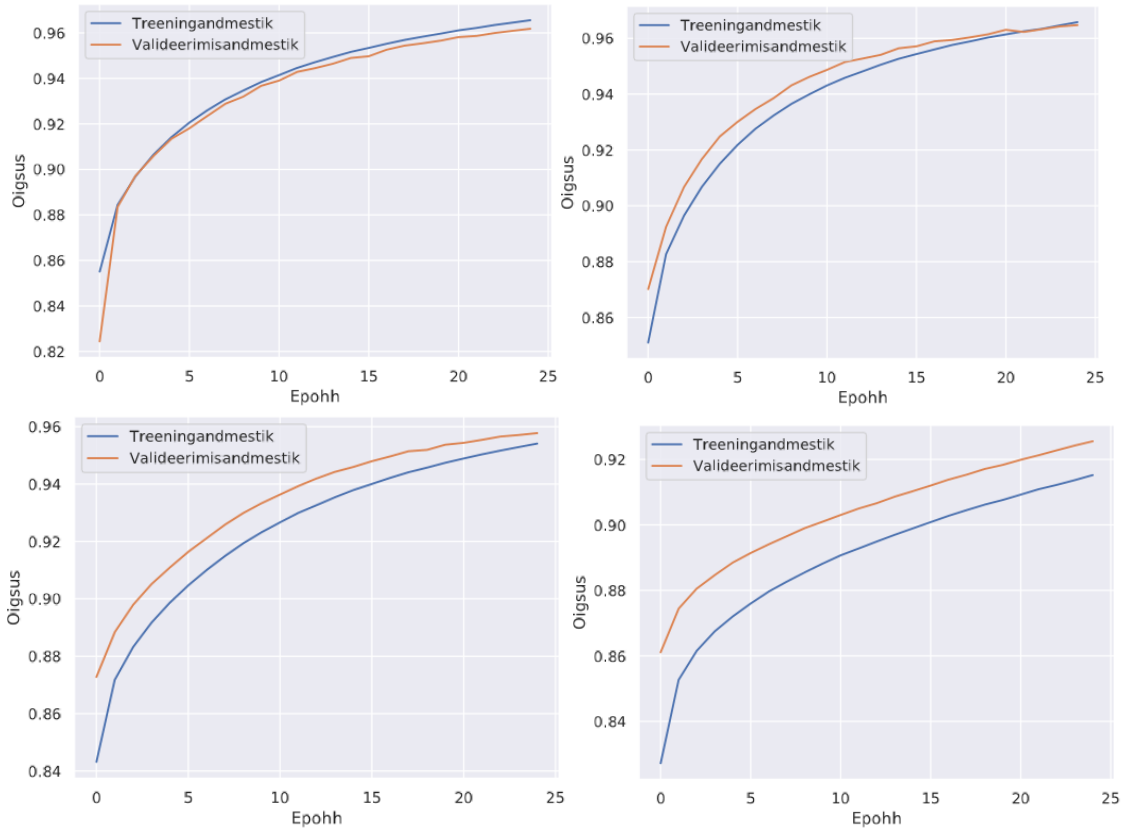
3.6.1. Õpisammu mõju NNa-le

NNa-s kasutatud Adagradi õpisamm on 0,002. **NNa_lr001** on NNa õpisammuga 0,001. **NNa_lr003** on NNa õpisammuga 0,003. **NNa_lr01** on NNa õpisammuga 0,01. Nende tulemused on toodud lisades 13–15.

Õigsuse muutuseid treenimisel treening- ja valideerimisandmestikul vaadates tundub kõige paremini hakkama saavat NNa õpisammuga 0,01, kuna sellel on õigsused mõlemal andmestikul väga sarnased ja kõrged. Jooniselt 9 on näha, et õpisammu vähenedes suureneb õigsuste vahe treening- ja valideerimisandmestikul. Samas on jooniselt 10 näha õpisammuga 0,01 NNa keskmist kahjut, mis viitab väikesele ülesobitamisele (kahju valideerimisandmestikul on kõrgem, kui treeningandmestikul). Teiste õpisammudega närvivõrkude keskmised kahjud ei sobita üle. Õpisammuga 0,003 NNa kahju jõuab madalamale, kuid õpisammuga 0,002 on kahjud veidi rohkem samal joonel. Õpisammuga 0,001 NNa kahju ei jõua nii madalale, kui eelneva kahe oma. Seega võib õigsuste ja kahjude graafikute põhjal öelda, et parim näib NNa õpisammuga 0,01, kuid sellele äratab kahtlust tema kahjude areng. Kõige parem näib olevat hoopiski mitte NNa (õpisammuga 0,002), vaid NNa_lr003 (õppisammuga 0,003), sest selle õigsused on üksteisel lähemal ja jõuavad kõrgemale ning kahjud madalamale.

Treenimisandmete ja keskmiste tulemuste tabelitest lisades 13–15 on näha õpisammuga 0,02 NNa ja õpisammuga 0,003 NNa sarnasust tulemustes: nii õpisammuga 0,01 NNa

kui ka õpisammuga 0,003 NNa keskmine õigsus testandmestikul on 0,93, MCC 0,48, kaalutud F1 0,95. Vaid NNa_lr01 kahju (0,25) on 0,01 võrra kõrgem õpisammuga 0,003 NNa-st (0,26) ja I tüübi viga (0,68) 0,01 võrra madalam (0,69). NNa ise tundub neid mõõdikuid vaadates NNa_lr01-st ja NNa_lr003-st ebatäpsem (testandmestikul keskmine õigsus 0,92, kahju 0,26, MCC 0,47, kaalutud F1 0,94, I tüübi viga 0,71). Õpisammuga 0,001 NNa eelnevate närvivõrkudega võistelda ei suuda: selle I tüübi viga (0,77) ja kahju (0,39) on kõigist suurem ja ülejäänud mõõdikud teistest madalamad (test-andmestikul keskmine õigsus 0,89, MCC 0,42, kaalutud F1 0,92).

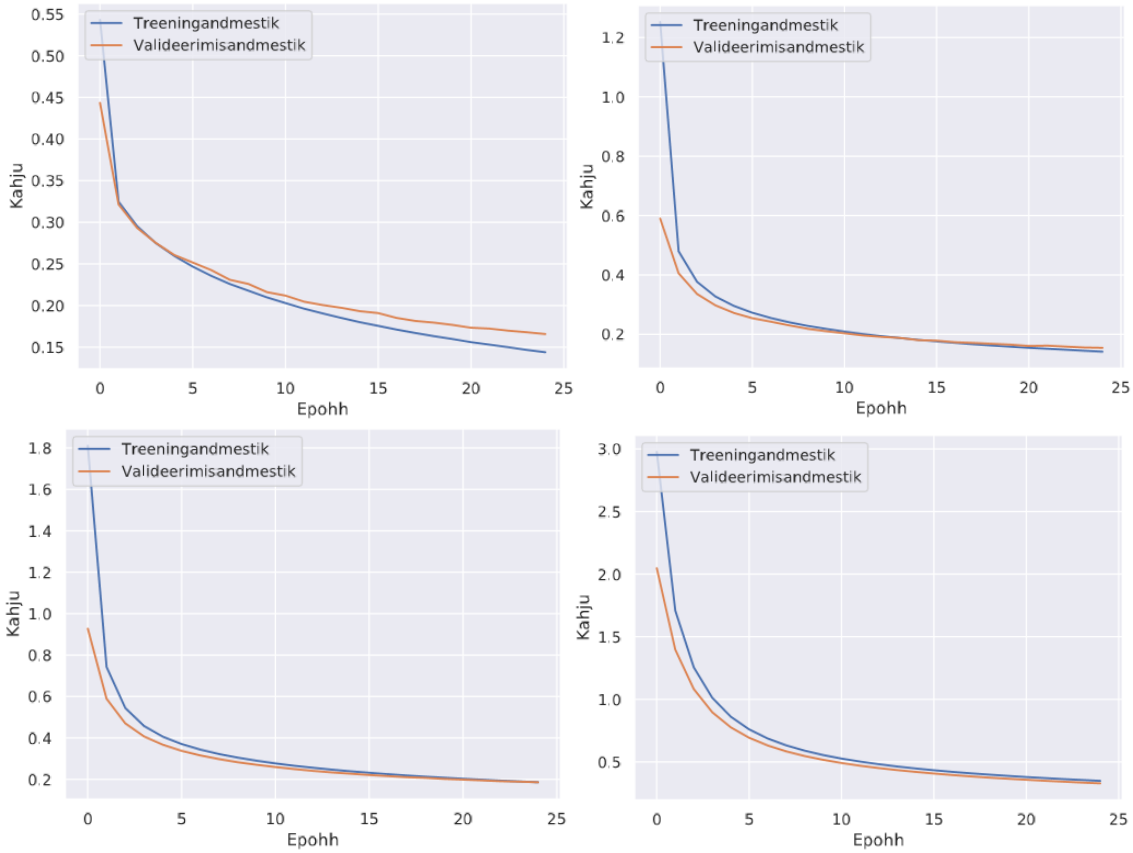


Joonis 9. **Erinevate õpisammudega NNa õigsuse muutused treening- ja valideerimisandmestikul epohhide lõikes. Üleval vasakult paremale: õpisamm on 0,01 ja 0,003. All vasakult paremale: õpisamm on 0,002 ja 0,001.**

Segadusmaatriksid (vt jooniselt 11) toovad parima närvivõrgu otsingutesse veidi rohkem selgust. Mida väiksem on õpisamm, seda suurem on viitsuhtes paaride saagis ja I tüübi viga ning seda väiksem on viitesuhteta paaride saagis. Kuna tulemused on arvutatud tasakaalust väljas testandmestiku ennustuste pealt, on viitesuhteta paaride leidmisel suurem mõju ka siis, kui mõõdikute valemites üritatakse vähendada osakaalu

erinevuste mõju ja sellest tulenevad ka väiksema õpisammuga NNa-de kõrgemad tulemused, kuid madalam viitesuhtes paaride tuvastusprotsent ehk saagis.

Nende andmete põhjal on näha, et siin töös eelistatud NNa on tegelikult üsna võrdne õpisammuga 0,003 treenitud NNa-ga. Kui eelistada madalamat viitesuhtes paaride saagist, aga saada see-eest juurde protsendi võrra kõrgem viitesuhteta paaride saagis koos madalama I tüübi veaga, tuleb valida parimaks NNa_lr003. Kuna töö eesmärk on leida parim viitesuhete lahendaja, eelistati selles töös peatükis 3.2 õpisammuga 0,002 NNa-d (õpisammuga 0,001 saavutatakse veel kõrgem viitesuhtes paaride saagis, kuid selle viitesuhteta paaride saagis langeb närvivõrgu eelistamiseks liiga madalale. Kusjuures, viimase segadusmaatriks on identne NNa_adasyn'i segadusmaatriksiga). Samas on kõikide selles alapeatükis mainitud närvivõrkude I tüübi viga kommertskasutuseks liiga kõrge.



Joonis 10. **Erinevate õpisammudega NNa kahju muutused treening- ja valideerimisandmestikul epohhide lõikes. Üleval vasakult paremale: õpisamm on 0,01 ja 0,003. All vasakult paremale: õpisamm on 0,002 ja 0,001.**

I tüübi viga	0,68	Mudeli ennustus	
II tüübi viga	0,01	Viite-suhtes	Viite-suhteta
Tegelikult	Viite-suhtes 100%	910,04 80%	224,96 20%
	Viite-suhteta 100%	1972,43 6%	29379,52 94%

I tüübi viga	0,69	Mudeli ennustus	
II tüübi viga	0,01	Viite-suhtes	Viite-suhteta
Tegelikult	Viite-suhtes 100%	933,11 82%	201,89 18%
	Viite-suhteta 100%	2039,23 7%	29312,77 93%

I tüübi viga	0,71	Mudeli ennustus	
II tüübi viga	0,01	Viite-suhtes	Viite-suhteta
Tegelikult	Viite-suhtes 100%	964,18 85%	170,82 15%
	Viite-suhteta 100%	2369,1 8%	28982,9 92%

I tüübi viga	0,77	Mudeli ennustus	
II tüübi viga	0,0	Viite-suhtes	Viite-suhteta
Tegelikult	Viite-suhtes 100%	1014,91 89%	120,09 11%
	Viite-suhteta 100%	3420,77 11%	27931,23 89%

Joonis 11. **Erinevate õpisammudega NNa keskmised segadusmaatriksid testandmestikul. Üleval vasakult paremale: õpisamm on 0,01 ja 0,003. All vasakult paremale: õpisamm on 0,002 ja 0,001.**

3.6.2. Õpisammu mõju NN1-le

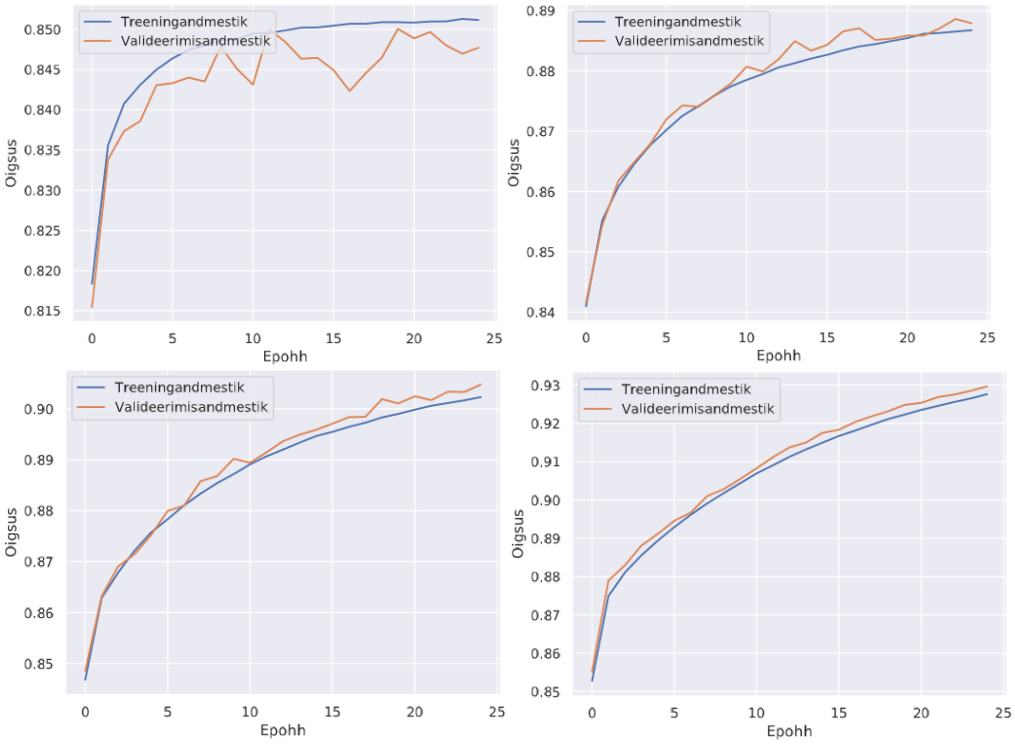
NN1-s kasutatud Adami õpisamm on 0,001. **NN1_lr01** on NN1 õpisammuga 0,01. **NN1_lr002** on NN1 õpisammuga 0,002. **NN1_lr003** NN1 on õpisammuga 0,003. Lisades 16–18 on toodud nende närvivõrkude tulemused.

Tulemusi vaadates on näha, et Adami õpisammu kasvades (0,001→0,002→0,003→0,01) kahaneb keskmine MCC (0,44→0,4→0,39→0,35), õigsus (0,91→0,88→0,87→0,84) ja kaalutud keskmine F1 (0,93→0,91→0,91→0,89) testandmestikul. Õpisammu kasvades tõuseb kahju (0,29→0,32→0,34→0,43) ja I tüübi viga (0,75→0,79→0,8→0,84). Nende tulemuste põhjal võib kinnitada, et madalaim NN1-e õpisamm (0,001) annab paremaid tulemusi.

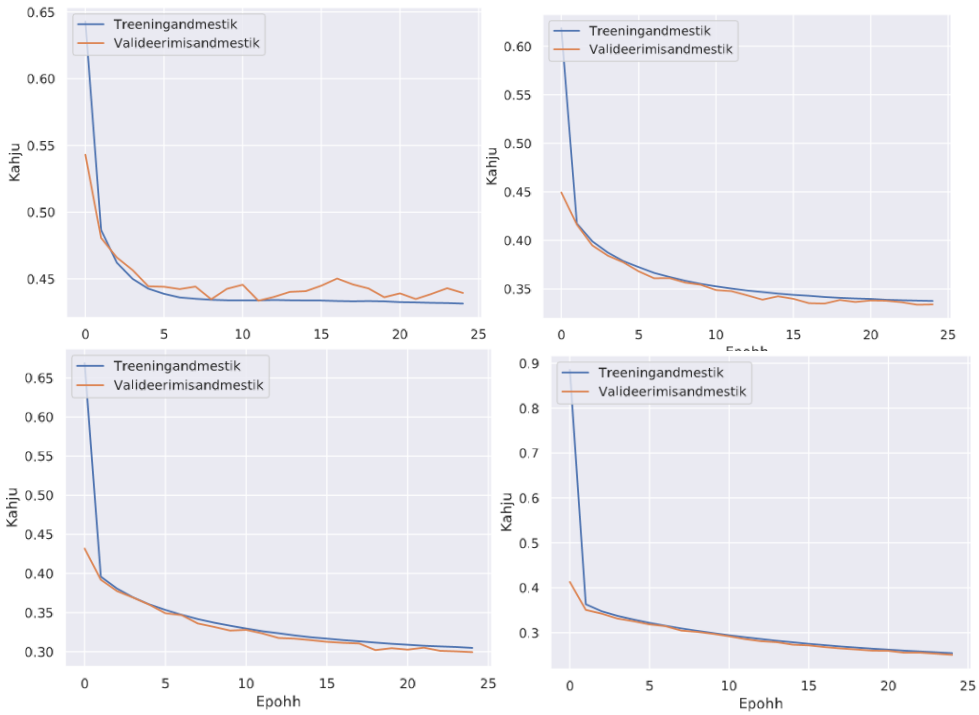
Huvitaval kombel määrab madalaima õpisammuga NN1 teistest halvemini viitesuhtes

paare õigesti (saagis on 85%). Ülejäänud närvivõrgud NN1_lr002, NN1_lr003 ja NN1_lr01 määravad viitesuhtes paaridest 88% õigesti. Tulemuste paremus tuleneb jällegi viitesuhteta paaride madalamast äratundmisest õpisammu suurenedes (vastavalt õpisammu suurenemisele: 91%→88%→87%→83%). NN1 õpisammu 0,001 võrra suurendamine (0,002-ni) tõstab 3% viitesuhtes paaride saagist, kuid langetab selle võrra 3% viitesuhteta paaride saagist. Seetõttu tõuseb ka I tüübi viga 0,04 võrra. Neid muutusi kaaludes otsustati siin magistritöös õpisammuga 0,001 NN1 kasuks. Samas võib eelistada ka õpisammu 0,002.

Seda otsust toetab ka kahju ja õigsuse analüüs treening- ja valideerimisandmestikul (vt jooniseid 12 ja 13), mis näitavad tulemuste hüppamist ja kõikumist õpisammu suurenedes.



Joonis 12. **Erinevate õpisammudega NN1 keskmised õigsuse muutused treening- ja valideerimisandmestikul. Üleval vasakult paremale: õpisamm on 0,01 ja 0,003. All vasakult paremale: õpisamm on 0,002 ja 0,001.**



Joonis 13. **Erinevate õpisammudega NN1 kahju muutused treening- ja valideerimisandmestikul epohhide lõikes. Üleval vasakult paremale: õpisamm on 0,01 ja 0,003. All vasakult paremale: õpisamm on 0,002 ja 0,001.**

3.7. Epohhide (ingl epoch*) arv*

Seni on kõik katsed tehtud 25 viie epohhiga (vt lk 40). Nii väikese andmestiku 25 korda läbi vaatamine ei pruugi olla mõistlik. See alapeatükk analüüsib NN1-e ja NNa tulemuste muutuseid, kui vähendada epohhide arvu treenimisel.

3.7.1. Epohhide arvu mõju NN1-le

NN1-s kasutati 25 epohhi. **NN1_epoch5** on NN1 treenitud 5 epohhiga, **NN1_epoch15** 15 epohhiga, **NN1_epoch20** 20 epohhiga. Epohhide kasvades (5→15→20→25) kasvab ka keskmine õigsus (0,87→0,9→0,9→0,91), MCC (0,39→0,42→0,43→0,44) ja kaalutud F1 (0,91→0,92→0,93→0,93) tasakaaluta testandmestikul. Kui väärtus jääb neil mõõdikutel eelmisega võrreldes samaks, siis on natukene kõrgenenud usaldusvahemik. Kahju langeb (0,35→0,3→0,3→0,29). Sama teeb ka I tüübi viga (0,8→0,77→0,76→0,75). (Vt täpsemalt lisadest 19–21) Seega võib nende mõõdikute põhjal öelda, et epohhide arvu suurendamine tõstab närvivõrgu tulemusi.

Kui aga vaadata segadusmaatrikseid jooniselt 14, on näha, et viitesuhtes paaride tuvastustäpsus langeb epohhide suurenemisel (89%→87%→86%→85%) ja närvivõrgu edu tuleneb viitesuhteta paaride tuvastustäpsuse suurenemisel (91%→92%→93%→ 93%) (suurim usaldusvahemik on viimasel).

Epohhi arvu valides tuleb sellega arvestada. Magistritöö edukaimaks närvivõrguks valiti 25 epohhiga närvivõrk. Selle tugevuseks on nende katsete suurim keskmine õigsus testandmestikul (tegelikult ka treening- ja valideerimisandmestikul) ja madalaim kahju. Ka keskmine MCC ja kaalutud keskmine F1 on katsete parimad. Kuigi selle närvivõrgu ennustatult viitesuhtes paaridest on kõrgeim protsent tegelikult ka viitesuhtes (siiski ainult 25%), tuleb selle närvivõrgu eelistamine oma hinnaga: madalaim protsent viitesuhtes paaridest määratakse õigesti (saagis 85%). Seega võib tegelikult kaaluda ka väiksemat epohhide arvu. Näiteks eelistades 20 epohhiga NN1_epoch20-t 25 epohhiga NN1 asemel, kaotatakse viitesuhteta paaride saagises ja I tüübi veas üks protsent, aga võidetakse protsent viitesuhtes paaride saagises.

I tüübi viga	0,8	**Mudeli ennustus**	
II tüübi viga	0,0	Viite-suhtes	Viite-suhteta
Tegelikult	Viite-suhtes 100%	1010,37 89%	124,63 11%
	Viite-suhteta 100%	4104,63 13%	27247,37 87%

I tüübi viga	0,77	**Mudeli ennustus**	
II tüübi viga	0,01	Viite-suhtes	Viite-suhteta
Tegelikult	Viite-suhtes 100%	984,09 87%	150,91 13%
	Viite-suhteta 100%	3222,77 10%	28129,23 90%

I tüübi viga	0,76	**Mudeli ennustus**	
II tüübi viga	0,01	Viite-suhtes	Viite-suhteta
Tegelikult	Viite-suhtes 100%	976,75 86%	158,25 14%
	Viite-suhteta 100%	3033,5 10%	28318,5 90%

I tüübi viga	0,75	**Mudeli ennustus**	
II tüübi viga	0,01	Viite-suhtes	Viite-suhteta
Tegelikult	Viite-suhtes 100%	968,15 85%	116,85 15%
	Viite-suhteta 100%	2842,81 9%	28509,19 91%

Joonis 14. **Erinevate epohhide arvuga treenitud NN1-e keskmised segadusmaatriksid tasakaaluta testandmestikul. Üleval vasakult paremale: epohh on 5 ja 15. All vasakult paremale: epohh on 20 ja 25.**

3.7.2. Epohhide arvu mõju NNa-le

NNa treeniti 25 epohhiga. **NNa_epoch5** on 5 epohhiga treenitud NN1, **NNa_epoch15** 15 epohhiga, **NNa_epoch20** 20 epohhiga. Huvitav on märkida, et närvivõrgu NNa_epoch5 segadusmaatriks (vt lisa 22) on võrdne närvivõrkude NNa_lr001 ja NNa_adasyn tulemiga. Samuti on siin näha Adagradi paremust Adami ees: 5 epohhiga jõuab Adagrad pea sama kaugele, kui Adam 15 epohhiga. Tõsi, viitesuhtes paaride tuvastusprotsent on Adagradil juba 5 epohhiga kõrgem, kui Adamil 15 epohhiga.

NNa tulemuste arengut epohhide lõikes uurides (vt lisasid 22–24) on näha sama tendentsi, mis NN1-lgi: epohhide kasvades (5→15→20→25) tõuseb treening-andmestikult võetud keskmine õigsus (0,89→0,91→0,92→0,92), MCC (0,41→0,45→ 0,46→0,47) ja kaalutud F1 (0,92→0,93→0,94→0,94). Viimase kahe õigsuse ja kaalutud F1 vahe on vaid kõrgemas ja madalamas usaldusvahemikes, ümardades saab sama tulemuse. Seetõttu on 20 ja 25 epohhiga treenitud NNa tulemused üsna ligilähedased.

Kahju (0,39→0,3→0,28→0,26) ja I tüübi viga (0,77→0,74→ 0,72→0,71) langevad. Samas segadusmaatriksitelt (vt lisasid 22–24) on näha, et paremad tulemused tulevad viitesuhtes paaride saagise langemise (89%→87%→86%→85%) ja viitesuhteta paaride saagise (89%→91%→92%→92%) suurenemise arvelt. Kuna 20 ja 25 epohhiga treenitud NNa tulemused on väga sarnased ning NNa_epoch20 tuvastab protsendi võrra paremini viitesuhtes paare, võib soovi korral seda eelistada NNa asemel. Seda eelistades läheb siiski MCC 0,01 võrra väiksemaks ning kahju ja I tüübi viga suuremaks.

3.8. Närvivõrkude võrdlus eelnevalt tehtud tööga

2020. aasta maikuus pole veel ilmunud Eduard Barbu katsete tulemused. Nagu eelnevalt mainitud (lk 5–6 ja 19), tegeles tema tunnuste eraldamise koodi poolega ja lõi töövoo, mis hakkab tundmatutes tekstides pronominaalseid viitesuhteid eraldama. Samuti proovis ta samal osutus-paari meetodil lahendada pronominaalseid viitesuhteid, kasutades mitte-närvivõrk masinõppe meetodeid.

Esialgsed tulemused näitavad, et parima tulemuse annab XGBoost. XGBoost (Chen,

Guestrin 2016) on otsustus-puu põhine (ingl *decision-tree-based*) edasi arenenud populaarne masinõppe algoritm ja sellel on ka samanimeline tarkvarateek, mida saab Pythonis kasutada (vt lähemalt XGBoosti dokumentatsioon). Selle viitesuhtes F1 on pea kogu treeningandmestikul treenides ja samamoodi tasakaalust väljas oleval testandmestikul testides 0,6. Meeldetuletusena – parima siin töös saavutatud närvivõrgu NNa viitesuhtes F1 on tasakaalus testandmestikul 0,43.

Kui NNa-d treenida kogu tasakaalust väljas oleval andmestikul (viitesuhtes paare 10 korda korrutamata ja juhuslikult sama palju viitesuhteta paare valimata) saab see närvivõrk samuti viitesuhtes F1 väärtuseks 0,6 (vt NNa_alldata tulemusi lisast 5 või analüüsi ptk 3.4). Mäletatavasti suudab see närvivõrk siiski õigesti märkida vaid pooled viitesuhtes paaridest ja seetõttu langes konkurentsist kohe välja. Analüüsis viitesuhtes F1-le eraldi tähelepanu ei pööratud, vaid vaadati erinevaid mõõdikuid korraga. Siinkohal võib välja tuua, et NNa-st parema viitesuhtes F1-ga on NNa_lr003 (0,46), NNa_1pos3neg (0,45), NNa_lr01 (0,46) ja sama viitesuhtes F1-ga on NNa_batch128 (0,43), NNa_batch64 (0,43), NNa_batch512 (0,43). Ka analüüsist selgus, et ploki suurus ei mõjuta tulemust, ning erineva õpisammuga NNa ning sisendiga 1pos3neg analüüsi ja valikute kohta saab lugeda peatükkidest 3.6.1 ja 3.4.

Kuna XGBoosti tulemuses on hetkel olemas vaid viitesuhtes F1 ja muu analüüs on pooleli, ei saa põhjapanevaid järeldusi veel teha ega eelistada üht meetodit teisele.

Puolakaineni ja Mutso reeglipõhised lähenemised keskendusid asesõnadele *tema* ja *nemad* ja lahendasid 70–79% ning alla 74% viitesuhetest. NNa lahendas asesõnu *kes, mis, mina, sina, tema, meie, teie, nemad* ning suutis lahendada 85% viitesuhetest.

KOKKUVÕTE

Magistritöö eesmärk oli luua närvivõrk, mis tuvastab automaatselt pronominaalseid viitesuhteid eesti keeles. Eesmärgist lähtuvalt tutvustab magistritöö viitesuhete lahendamisega seotud tähtsamaid mõisteid, annab lühikese ülevaate viitesuhete lahendamise arengust, tutvustab erinevaid lahendamismeetodeid ja keskendub närvivõrkudele viitesuhete automaatse lahendamise kontekstis.

Töös lahendatakse viitesuhteid osutus-paari meetodil, kus närvivõrgu sisend on 32 tunnusega vektor pronoomeni ja tema võimaliku viitealuse (lähedal oleva pronoomeni, nimisõna või pärisnime) paari kohta. Närvivõrk määrab nende tunnuste põhjal, kas tegu on viitesuhtes sõnade paariga või mitte. Viitealuseid otsiti küsiv-siduvatele asesõnadele *kes* ja *mis* ning isikulistele asesõnadele *mina, sina, tema, meie, teie, nemad*. Tunnused on eraldatud asendussõnade suhtes käsitsi märgendatud eesti ajalehtede korpusest, mille loomele ja suurendamisele aitas töö autor kaasa. Tunnuste kirjeldust saab lugeda peatükis 2.3. Korpuse kirjeldust ja eeltöötluse kohta saab lugeda peatükis 2.2.

Närvivõrkude analüüsis peatükis 3 tutvustati eksperimenteerimise käigus leitud parimaid närvivõrke ja analüüsiti erinevate parameetrite mõju närvivõrkude tulemustele. Analüüsis selgus, et laiem närvivõrk (rohkemate sõlmedega) annab paremaid tulemusi, arvandmeliste tunnuste min-max normaliseerimine *1-hot-encoding*'u asemel ei tõsta närvivõrgu kvaliteeti, ADASYN'i kasutamine ei tõstnud tulemusi ning ploki suuruse muutmisel ei olnud mingit erilist mõju tulemustele. Samuti leiti, et epohhide arvu suurendamine tõstab viitesuhteta paaride saagist, kuid alandab viitesuhtes paaride saagist. Sama saab öelda ka Adagradi õpisammu kohta: mida väiksem on õpisamm, seda suurem on õigesti ennustatud viitesuhtes paaride osakaal ja I tüübi viga ning seda väiksem on õigesti ennustatud viitesuhteta paaride osakaal (ehk saagis). Adami puhul annab õpisammu vähendamine väiksema I tüübi vea ja kõrgema viitesuhteta paaride saagise. Viitesuhtes paaride saagis muutub vähem, kuid pigem alaneb.

Analüüs illustreerib ka parima närvivõrgu parameetrite valimise keerukust – väike muutus võib tõsta mõnda olulist mõõdikut, kuid samas langetada mõnda teist olulist

mõõdikut. Seetõttu on oluline vaadelda tulemusi erinevate mõõdikute suhtes (loe nende kohta ptk 3.1) ja otsustada, millised neist on olulisemad.

Autori hinnangul töö eesmärk täideti. Kõikide töös tehtud katsete koodid on üleval githubis: https://github.com/Lindafr/EstPronCorefNN. Edukaim närvivõrk on viie-kihiline Adagradiga närvivõrk (githubis nimega NNa), milles kasutatakse väljajätumeetodit, L2-regulariseerimist ja ploki normaliseerimist. See tuvastab 85% viitesuhtes paaridest ja 91% viitesuhteta paaridest. Selle õigsus on 0,92, kahju 0,26 ja MCC 0,47 testandmestikul. Viitesuhtes F1 on 0,43, kaalutud keskmine F1 on 0,94. Halvimaks näitajaks on I tüübi viga (0,71).

Tegu on vaid ühe sammuga viitesuhete lahendamisel eesti keeles. Selle teema edendamiseks soovitab autor uuendada asendussõnade suhtes käsitsi märgendatud korpust nii, et kõik viitesuhted ja -ahelad oleksid seal märgendatud. See lubab katsetada muid meetodeid, mis keskenduvad tervetele viiteahelate lahendamisele, (vt peatükki 1) ning lahendada peale pronoomenite ka kõiki muid samale olemile viitavaid sõnu.

Närvivõrgu edasi arendamiseks soovitab autor uurida ka närvivõrgu tulemusi pronoomenite kaupa (millised pronoomenid see kõige paremini ära tunneb?) ning samuti suurendada korpust ning lisada tunnuseid (masinõppes on rohkem andmeid alati parem). Närvivõrgu parameetrite paremaks leidmiseks võib proovida ka automaatseid meetodeid nagu võreotsingut (ingl *grid search*) ja juhuslikku otsingut (ingl *random search*) (Goodfellow 2016: 422–431).

KIRJANDUS

Abstractness_ET = Aedmaa, Eleri 2019. Eleri Aedmaa doktoritöö käigus valminud abstraktsuse skoori leksikon eesti keeles. https://github.com/eleriaedmaa/compositionality/tree/master/datasets/abstractness_ET. Vaadatud 21.05.2020.

Aedmaa, Eleri 2019. Detecting compositionality of Estonian particle verbs with statistical and linguistic methods. Doktoritöö. Tartu Ülikool. Humanitaarteaduste ja kunstide valdkond. https://dspace.ut.ee/handle/10062/65146. Vaadatud 09.03.2020.

Banerjee jt = Banerjee, A., Chitnis, U. B., Jadhav, S. L., Bhawalkar, J. S., Chaudhury, S. 2009. Hypothesis testing, type I and type II errors. Industrial Psychiatry Journal, kd 18, nr 2, lk 127–131. https://doi.org/10.4103/0972-6748.62274. Vaadatud 11.03.2020.

Charniak, E., Elsner, M. 2009. EM works for pronoun anaphora resolution. Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics. Athens, Greece: Association for Computational Linguistics. lk 148–156. (EACL 2009). https://www.aclweb.org/anthology/E09-1018. Vaadatud 01.02.2020.

Chicco, D., Jurman, G. 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 21, 6. https://doi.org/10.1186/s12864-019-6413-7. Vaadatud 11.03.2020.

Charniak, E., Elsner, M. 2009. EM works for pronoun anaphora resolution. Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics. Ateena, Kreeka: Association for Computational Linguistics. lk 148–156. (EACL ’09). https://www.aclweb.org/anthology/E09-1018/. Vaadatud 22.01.2020.

Chen, T., Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, lk 785–794.

Clevert jt = Clevert, Djork-Arné, Unterthiner, T., Hochreiter, S. 2016. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). Konverentsi ICLR 2016 paber. https://arxiv.org/abs/1511.07289. Vaadatud 10.03.2020.

Devlin jt = Devlin, Jacob, Chang, M.-W., Lee, K., Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Konverentsi NAACL-HLT 2019 paber. Minnesota. Lk 4171–4186. https://arxiv.org/abs/1810.04805. Vaadatud 24.01.2020.

Duchi jt = Duchi, John, Hazan, E., Singer, Y. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research, kd 12, lk 2121–2159.

Durrett, G., Klein, D. 2013. Easy Victories and Uphill Battles in Coreference Resolution. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle, Washington, USA: Association for Computational Linguistics. lk 1971–1982. https://www.aclweb.org/anthology/D13-1203. Vaadatud 21.01.2020.

Eesti keele universaalse süntaksi vahendid ja rakendused. Projekti info Eesti Teadusinfosüsteemis. https://www.etis.ee/Portal/Projects/Display/05f643b4-431d-4e52-a727-d1ca7fb71cff. Vaadatud 21.05.2020.

Entu = Eesti Keeleressursside Keskuse sõnavektorid. https://entu.keeleressursid.ee/public-document/entity-7540/word-embeddings. Vaadatud 09.03.2020.

Fernandes jt = Fernandes, Eraldo, dos Santos, C., Milidiú, R. 2012. Latent Structure Perceptron with Feature Induction for Unrestricted Coreference Resolution. Joint Conference on EMNLP and CoNLL. Shared Task. Jeju saar, Korea: Association for Computational Linguistics. lk 41–48. https://www.aclweb.org/anthology/W12-4502. Vaadatud 21.01.2020.

Freienthal, Linda 2018. Pronominaalsete viitesuhete analüüs asendussõnade suhtes käsitsi märgendatud korpuses. Bakalaureusetöö. Juhendaja Kadri Muischnek. Tartu Ülikool.

Godoy, Daniel 2019. Understanding binary cross-entropy / log loss: a visual explanation. Towards Data Science. https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a. Vaadatud 11.03.2020.

Goodfellow jt = Goodfellow, Ian, Bengio, Y., Courville, A. 2016. Deep Learning. MIT Press. http://www.deeplearningbook.org/. Vaadatud 10.03.2020.

Haibo He jt = Haibo He, Yang Bai, Garcia, E. A., Shutao Li 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). lk 1322–1328.

Hinton jt = Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. R. 2012. Improving neural networks by preventing co-adaptation of feature detectors. Toronto ülikool. https://arxiv.org/abs/1207.0580. Vaadatud 10.03.2020.

Ioffe, S., Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Proceedings of the 32nd International Conference on International Conference on Machine Learning, kdt 37, lk 448–456. https://arxiv.org/abs/1502.03167. Vaadatud 10.03.2020.

Joshi jt = Joshi, Mandar, Levy, O., Zettlemoyer, L., Weld, D. 2019. BERT for Coreference Resolution: Baselines and Analysis. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, Hiina: Association for Computational Linguistics. lk 5803–5808. https://www.aclweb.org/anthology/D19-1588. Vaadatud 24.01.2020.

Jurafsky, D., Martin, J. H. 2019. Coreference Resolution. 22. peatüki mustand raamatust Speech and Language Processing. https://web.stanford.edu/~jurafsky/slp3/. Vaadatud 22.01.2020.

Kingma, D. P., Ba, J. 2017. Adam: A Method for Stochastic Optimization. Kolmanda International Conference for Learning Representations paber. San Diego 2015. https://arxiv.org/abs/1412.6980. Vaadatud 10.03.2020.

Lappin, Shalom 2005. A Sequenced Model of Anaphora and Ellipsis Resolution.

Current Issues in Linguistic Theory. Toim António Branco, Tony McEnery, Ruslan Mitkov. Amsterdam: John Benjamins Publishing Company, lk 3. https://doi.org/10.1075/cilt.263.03lap. Vaadatud 22.01.2020.

Luo, X., Pradhan, S. 2016. Evaluation Metrics. Raamatus Anaphora Resolution. Algorithms, Resources, and Applications. Toim. Massimo Poesio, Roland Stuckardt, Yannick Versley. Theory and Applications of Natural Language Processing. Edited volumes. Lk 141–163. Springer.

Kenton Lee jt = Lee, Kenton, He, L., Lewis, M., Zettlemoyer, L. 2017. End-to-end Neural Coreference Resolution. Konverentsi Empiirilised Meetodid Loomuliku Keele Töötluses 2017 (EMNLP 2017) paber. https://arxiv.org/abs/1707.07045. Vaadatud 24.01.2020.

Lee, K. jt = Lee, Kenton, He, L., Zettlemoyer, L. 2018. Higher-Order Coreference Resolution with Coarse-to-Fine Inference. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). New Orleans, Louisiana: Association for Computational Linguistics. lk 687–692. https://www.aclweb.org/anthology/N18-2108. Vaadatud 24.01.2020.

Lee, H. jt = Lee, Heeyoung, Surdeanu, M., Jurafsky, D. 2017. A scaffolding approach to coreference resolution integrating statistical and rule-based models. Natural Language Engineering, kd 23, nr 05, lk 733–762. https://doi.org/10.1017/S1351324917000109. Vaadatud 20.01.2020.

Mitkov, Ruslan 2002. Anaphora Resolution. Suurbritannia. Pearson Education.

Miyato jt = Miyato, Takeru, Dai, A. M., Goodfellow, I. 2017. Adversarial Training Methods for Semi-Supervised Text Classification. Konverentsi ICLR 2017 paber. https://arxiv.org/abs/1605.07725. Vaadatud 26.01.2020.

Muischnek, K., Müürisep, K. 2016. Eesti keele sõltuvuspuude pank ja selle keeleteoreetilised lähted. Emakeele Seltsi aastaraamat, kd 62, nr 1, lk 122.

Muischnek, K., Müürisep, K. 2017. Estonian Copular and Existential Constructions as an UD Annotation Problem. Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017). Gothenburg, Rootsi: Association for Computational Linguistics. lk 79–85. https://www.aclweb.org/anthology/W17-0410. Vaadatud 03.02.2020.

Mutso, Pilleriin 2008. Knowledge-poor Anaphora Resolution System for Estonian. Magistritöö. Juhendaja Kaili Müürisep. Tartu Ülikool.

Ng coursera kursus = Ng, Andrew, Katanforoosh, K., Mourri, Y. B. 2020. Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization. Coursera. https://www.coursera.org/learn/deep-neural-network. Vaadatud: 10.03.2020.

Ng, Vincent 2017. Machine Learning for Entity Coreference Resolution: A Retrospective Look at Two Decades of Research. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17).

Ng, V., Cardie, C. 2002. Improving machine learning approaches to coreference resolution. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Philadelphia, Pennsylvania: Association for

Computational Linguistics. lk 104–111. (ACL ’02). https://doi.org/10.3115/1073083.1073102. Vaadatud 26.01.2020.

Novák, Michal 2018. Coreference from the Cross-lingual Perspective. Studies in computational and theoretical linguistics. Institute of Formal and Applied Linguistics. Doktoritöö. Juhendaja Zdeněk Žabokrtský. Charles University. Praha

OntoNotes = OntoNotes Release 5.0 lehekülg. https://catalog.ldc.upenn.edu/LDC2013T19. Vaadatud 22.05.2020

Pedregosa jt = Pedregosa, Fabian, Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., jt 2011. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, kd 12, nr 85, lk 2825–2830.

Pennington jt = Pennington, Jeffrey, Socher, R., Manning, C. 2014. Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics. lk 1532–1543. http://aclweb.org/anthology/D14-1162. Vaadatud 24.01.2020.

Peters jt = Peters, Matthew E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L. 2018. Deep contextualized word representations. Paber konverentsil NAACL HLT 2018. New Orleans. https://arxiv.org/abs/1802.05365. Vaadatud 24.01.2020.

Prechelt, Lutz 2012. Early Stopping — But When? Neural Networks: Tricks of the Trade: Second Edition. Toim Grégoire Montavon, Geneviève B. Orr, Klaus-Robert Müller. (Lecture Notes in Computer Science) Berlin, Heidelberg: Springer, lk 53–67.

Puolakainen, Tiina 2015. Anaphora resolution experiment with CG rules. Konverentsi NODALIDA 2015 töötoa artikkel. https://www.ep.liu.se/en/conference-article.aspx?series=ecp&issue=113&Article_No=6. Vaadatud 20.01.2020.

Scikit-learn’i kasutusjuhendi andmete eeltöötlemise peatükk. https://scikit-learn.org/stable/modules/preprocessing.html. Vaadatud 02.05.2020.

Sapena jt = Sapena, Emili, Padró, L., Turmo, J. 2013. A Constraint-Based Hypergraph Partitioning Approach to Coreference Resolution. Computational Linguistics, kd 39, nr 4, lk 847–884. https://www.aclweb.org/anthology/J13-4003/. Vaadatud 21.01.2020.

Sihipärane süntaks korpuste jaoks. Eesti keeletehnoloogia lehekülg. https://www.keeletehnoloogia.ee/et/ekt-projektid/sihiparane-suntaks-korpuste-jaoks. Vaadatud 01.02.2020.

Song jt = Song, Yang, Jiang, J., Zhao, X., Li, S., Wang, H. 2012. Joint Learning for Coreference Resolution with Markov Logic. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Jeju saar, Korea: Association for Computational Linguistics. https://www.aclweb.org/anthology/D12-1114. Vaadatud 22.01.2020.

Soon jt = Soon, Wee Meng, Ng, H. T., Lim, D. C. Y. 2001. A Machine Learning Approach to Coreference Resolution of Noun Phrases. Computational

Linguistics, kd 27, nr 4, lk 521–544. https://www.aclweb.org/anthology/J01-4004/. Vaadatud 21.01.2020.

Stuckardt, R. 2016. Introduction. Raamatus Anaphora Resolution. Algorithms, Resources, and Applications. Toim. Massimo Poesio, Roland Stuckardt, Yannick Versley. Theory and Applications of Natural Language Processing. Edited volumes. Springer.

Stöttner, Timo 2019. Why Data should be Normalized before Training a Neural Network. Towards Data Science. https://towardsdatascience.com/why-data-should-be-normalized-before-training-a-neural-network-c626b7f66c7d. Vaadatud 01.05.2020.

Stylianou, N., Vlahavas, I. 2019. A Neural Entity Coreference Resolution Review. https://arxiv.org/abs/1910.09329. Vaadatud 23.01.2020.

Subramanian, S., Roth, D. 2019. Improving Generalization in Coreference Resolution via Adversarial Training. Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019). Minneapolis, Minnesota: Association for Computational Linguistics. lk 192–197. https://www.aclweb.org/anthology/S19-1021. Vaadatud 26.01.2020.

Sundheim, B. M. 1995. Overview of Results of the MUC-6 Evaluation. Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995. https://www.aclweb.org/anthology/M95-1002. Vaadatud 20.01.2020.

Zhang jt = Zhang, H., Song, Y., Song, Y. 2019. Incorporating Context and External Knowledge for Pronoun Coreference Resolution. 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). Minneapolis, Minnesota: Association for Computational Linguistics. lk 872–881. https://www.aclweb.org/anthology/N19-1093. Vaadatud 03.02.2020.

Turian jt = Turian, Joseph, Ratinov, L.-A., Bengio, Y. 2010. Word Representations: A Simple and General Method for Semi-Supervised Learning. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Uppsala, Rootsi: Association for Computational Linguistics. lk 384–394. https://www.aclweb.org/anthology/P10-1040. Vaadatud 24.01.2020.

UD = Universal Dependencies formaati tutvustav koduleheküIg. https://universaldependencies.org/. Vaadatud 21.05.2020.

Vicedo, J. L., Ferrández, A. 2000. Importance of Pronominal Anaphora Resolution in Question Answering Systems. Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics. Hong Kong: Association for Computational Linguistics. lk 555–562. https://www.aclweb.org/anthology/P00-1070. Vaadatud 19.05.2020.

Wikidata = Wikidata koduleheküIg. https://www.wikidata.org/wiki /Wikidata:Main_Page. Vaadatud 22.05.2020

WordNet = ingliskeelse WordNeti koduleheküIg. https://wordnet.princeton.edu/. Vaadatud 22.05.2020.

XGBoosti dokumentatsioon. https://xgboost.readthedocs.io/en/latest/index.html. Vaadatud 19.05.2020.

PRONOMINAL COREFERENCE RESOLUTION IN ESTONIAN WITH NEURAL NETWORKS

SUMMARY

Coreference resolution has been one of the core tasks in natural language processing and computational linguistics. This master's thesis gives a short overview of the history of it and describes various methods of solving the problem. It presents a five-layer neural network based on a mention-pair model. The input for the network is taken from a corpus manually annotated with pronominal coreference relations (contains ca 147 000 strings). From that 32 features about pronouns *kes, mis, mina, sina, tema, meie, teie* and *nemad* and their (possible) antecedents are extracted. The candidate for the antecedent can be a noun, proper noun, or pronoun.

The test set contains 1135 coreference pairs and 31352 non-coreference pairs. The best network predicts 85% of coreference pairs and 92% of non-coreference pairs correctly on the test set. The accuracy is 92% and MCC is 0.47. The binary F1 is 0.43 and weighted average F1 is 0,94. Binary cross-entropy loss is 0.26 and type I error is 71%.

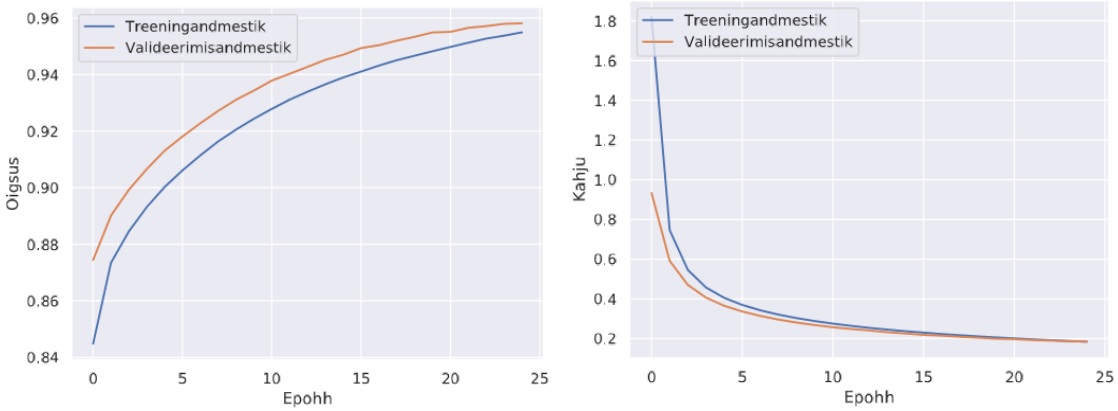
LISA 1. NN1 JA NNA TULEMUSED TASAKAALUS TESTANDMESTIKUL

Tabel 1.1. **NNa treenimisandmed ja keskmised tulemused tasakaalus testandmestikul.**

Treenimis-andmed	**NNa**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,89 0,884–0,888	0,42 0,407–0,423	0,78 0,773–0,78
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 1135	Viitesuhteta	0,85 0,846–0,855	0,94 0,938–0,94	0,89 0,89–0,892
Treeniti x korda k-korda	x = 8 k = 7	Viitesuhtes	0,93 0,929–0,934	0,83 0,829–0,839	0,88 0,877–0,881
Ploki suurus	256	Makro keskmine	0.89 0,889–0,891	0,89 0,885–0,889	0,89 0,884–0,889
Epohhe	25	Kaalutud keskmine	0,89 0,889–0,891	0,89 0,885–0,889	0,89 0,884–0,889

Tabel 1.2. **NNa keskmine segadusmaatriks tasakaalus testandmestikul.**

I tüübi viga	**0,07**	**Mudeli ennustus**	
II tüübi viga	**0,15**	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	946,66 83%	188,34 17%
	Viitesuhteta 100%	70,27 6%	1064,73 94%



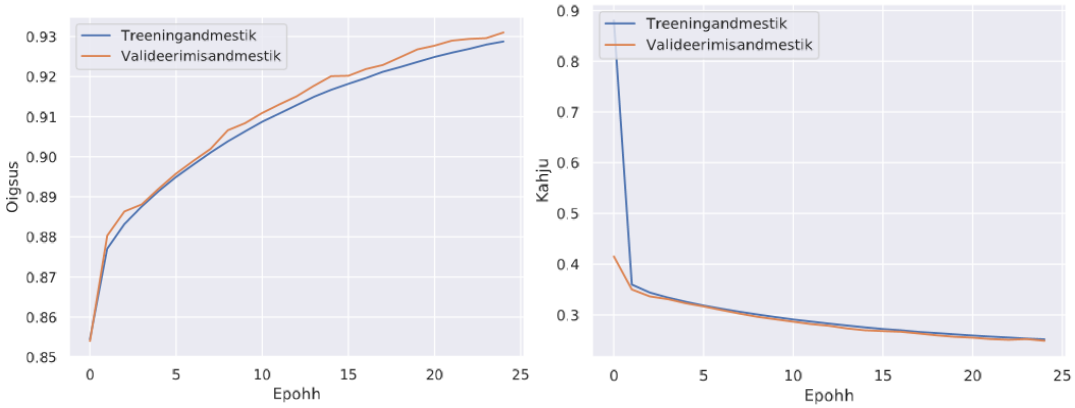
Joonis 1.1. **NNa õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

Tabel 1.3. **NN1 treenimisandmed ja keskmised tulemused tasakaalus testandmestikul.**

Treenimis-andmed	**NN1**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,88 0,88–0,884	0,42 0,41–0,427	0,77 0,763–0,77
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 1135	Viitesuhteta	0,86 0,849–0,862	0,92 0,913–0,926	0,89 0,884–0,888
Treeniti x korda k-korda	x = 8 k = 7	Viitesuhtes	0,91 0,91–0,919	0,84 0,835–0,852	0,88 0,874–0,879
Ploki suurus	256	Makro keskmine	0,89 0,883–0,887	0,88 0,881–0,884	0,88 0,88–0,884
Epohhe	25	Kaalutud keskmine	0.89 0,883–0,887	0.88 0,881–0,884	0.88 0,88–0,884

Tabel 1.4. **NN1 keskmine segadusmaatriks tasakaalus testandmestikul.**

I tüübi viga	**0,09**	**Mudeli ennustus**	
II tüübi viga	**0,15**	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	957,63 84%	177,38 16%
	Viitesuhteta 100%	90,89 8%	1044,11 92%



Joonis 1.2. **NN1 õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

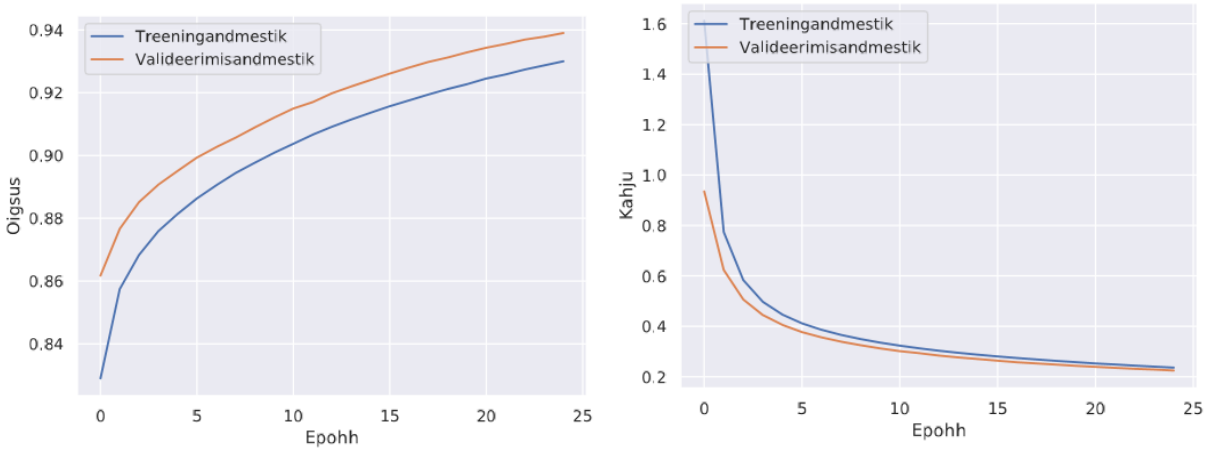
LISA 2. NNA_MINMAX TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 2.2. **NNa_minmax treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NNa_minmax**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51000	väärtus: usaldusvahemik:	0,9 0,896–0,9	0,3 0,295–0,303	0,43 0,424–0,43
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	1,0 0,995–0,998	0,9 0,897–0,9	0,94 0,943–0,945
Treeniti x korda k-korda	x = 10 k = 7	Viitesuhtes	0,24 0,237–0,243	0,88 0,875–0,881	0,38 0,373–0,38
Ploki suurus	256	Makro keskmine	0,62 0,616–0,619	0,89 0,888–0,89	0,66 0,658–0,662
Epohhe	25	Kaalutud keskmine	0,97 0,97–0,97	0,9 0,896–0,9	0,92 0,923–0,925

Tabel 2.2. **NNa_minmax keskmine segadusmaatriks.**

I tüübi viga	0,76	**Mudeli ennustus**	
II tüübi viga	0,0	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	996,44 88%	138,56 12%
	Viitesuhteta 100%	3171,03 10%	28180,97 90%



Joonis 2.1. **NNa_minmax õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

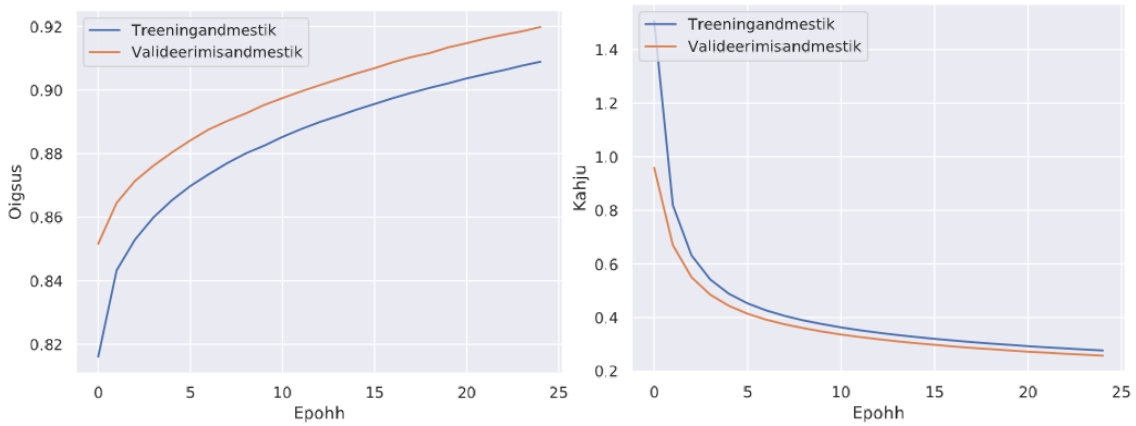
LISA 3. NN_MINMAX TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 3.3. **NN_minmax treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NN_minmax**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51000	väärtus: usaldusvahemik:	0,88 0,883–0,885	0,32 0,313–0,319	0,41 0,405–0,409
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	1,0 1,0–1,0	0,88 0,883–0,885	0,94 0,936–0,938
Treeniti x korda k-korda	x = 10 k = 7	Viitesuhtes	0,22 0,.216–0,22	0,89 0,892–0,894	0,35 0,348–0,352
Ploki suurus	256	Makro keskmine	0,61 0,605–0,608	0,89 0,889–0,89	0,64 0,641–0,644
Epohhe	25	Kaalutud keskmine	0,97 0,97–0,97	0,88 0,883–0,886	0,92 0,914–0,917

Tabel 3.2. **NN_minmax keskmine segadusmaatriks.**

I tüübi viga	0,78	**Mudeli ennustus**	
II tüübi viga	0,0	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	1013,81 89%	121,19 11%
	Viitesuhteta 100%	3649,43 12%	27702,57 88%



Joonis 3.1. **NN_minmax õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

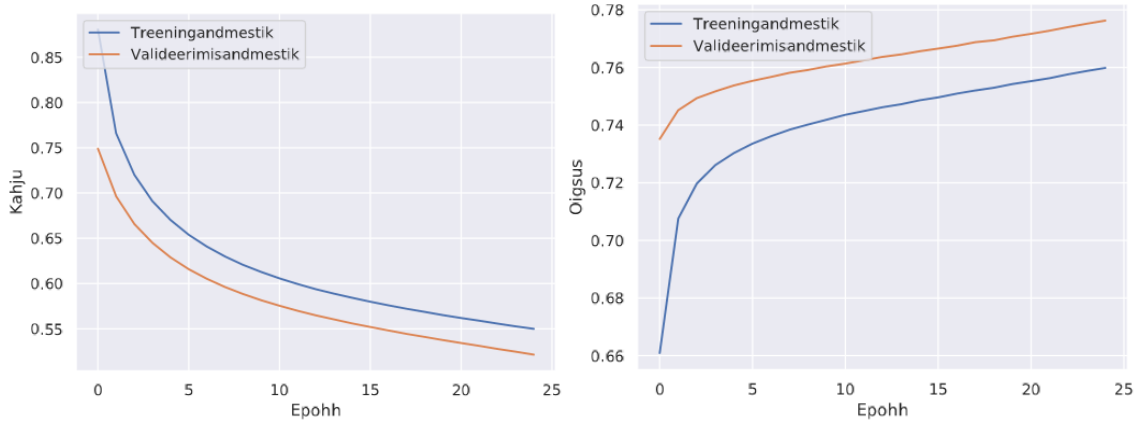
LISA 4. NN_NOCODING TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 4.4. **NN_nocoding treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NN_ nocoding**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,76 0,761–0,765	0,48 0,481–0,486	0,26 0,256–0,259
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	0,99 0,99–0,99	0,76 0,757–0,761	0,86 0,859–0,862
Treeniti x korda k-korda	x = 10 k = 7	Viitesuhtes	0,11 0,112–0,115	0,86 0,854–0,858	0,2 0,2–0,203
Ploki suurus	256	Makro keskmine	0,55 0,551–0,553	0,81 0,807–0,809	0,531 0,53–0,532
Epohhe	25	Kaalutud keskmine	0.96 0,96–0,96	0.76 0,761–0,766	0,84 0,836–0,839

Tabel 4.2. **NN_nocoding keskmine segadusmaatriks.**

I tüübi viga	0,89	**Mudeli ennustus**	
II tüübi viga	0,01	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	971,61 86%	163,39 14%
	Viitesuhteta 100%	7541,43 24%	23810,57 76%



Joonis 4.1. **NN_nocoding õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

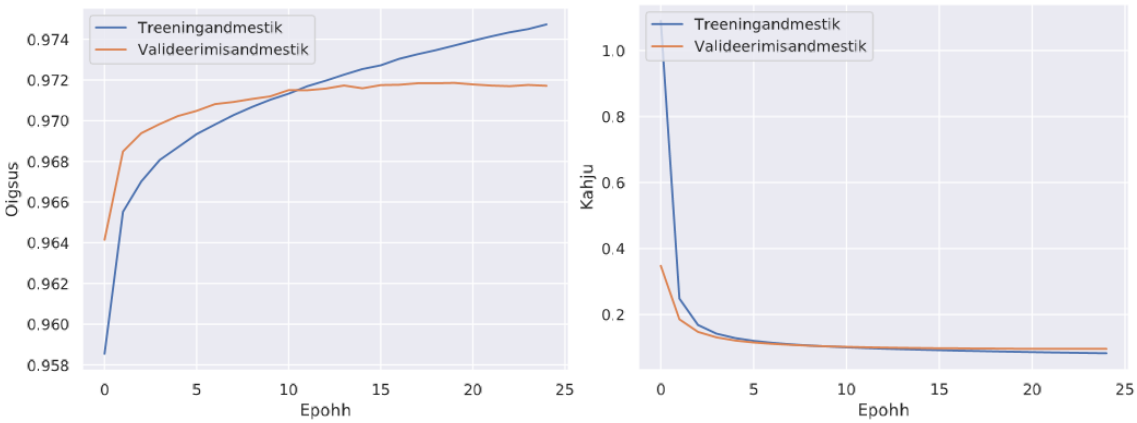
LISA 5. NNA_ALLDATA TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 5.5. **NNa_alldata treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NNa_ alldata**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 5095 0: 123846	väärtus: usaldusvahemik:	0,98 0,976–0,976	0,08 0,082–0,083	0,59 0,593–0,595
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	0,98 0,98–0,98	0,99 0,99–0,991	0,99 0,99–0,99
Treeniti x korda k-korda	x = 9 k = 12	Viitesuhtes	0,73 0,723–0,733	0,51 0,5–0,509	0,6 0,593–0,597
Ploki suurus	256	Makro keskmine	0,86 0,853–0,858	0,75 0,747–0,752	0,79 0,791–0,792
Epohhe	25	Kaalutud keskmine	0,97 0,97–0,97	0,98 0,978–0,98	0,97 0,97–0,97

Tabel 5.2. **NNa_alldata keskmine segadusmaatriks.**

I tüübi viga	0,27	**Mudeli ennustus**	
II tüübi viga	0,02	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	572,92 **50%**	562,08 **50%**
	Viitesuhteta 100%	216,18 1%	31135,82 99%



Joonis 5.1. **NNa_alldata õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

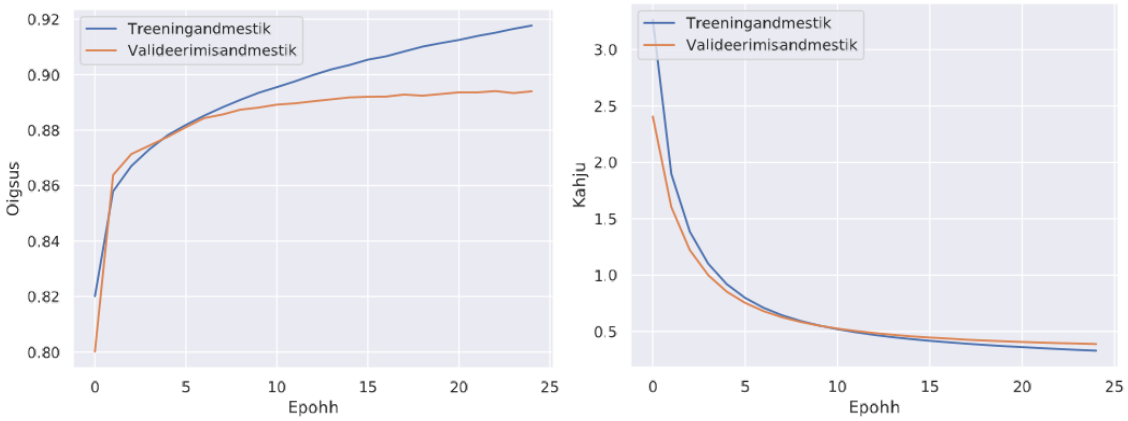
LISA 6. NNA_1POS3NEG TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 6.6. **NNa_1pos3neg treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NNa_ 1pos3neg**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 5095 0: 15285	väärtus: usaldusvahemik:	0,93 0,929–0,931	0,3 0,295–0,302	0,48 0,473–0,479
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	0,99 0,99–0,99	0,93 0,933–0,935	0,96 0,961–0,963
Treeniti x korda k-korda	x = 14 k = 8	Viitesuhtes	0,31 0,307–0,313	0,81 0,811–0,816	0,45 0,445–0,452
Ploki suurus	256	Makro keskmine	0,65 0,65–0,654	0,87 0,873–0,875	0,71 0,703–0,708
Epohhe	25	Kaalutud keskmine	0,97 0,97–0,97	0,93 0,929–0,931	0,94 0,943–0,945

Tabel 6.2. **NNa_1pos3neg keskmine segadusmaatriks.**

I tüübi viga	0,69	**Mudeli ennustus**	
II tüübi viga	0,01	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	924,08 81%	210,92 19%
	Viitesuhteta 100%	2070,08 7%	29281,92 93%



Joonis 6.1. **NNa_1pos3neg õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

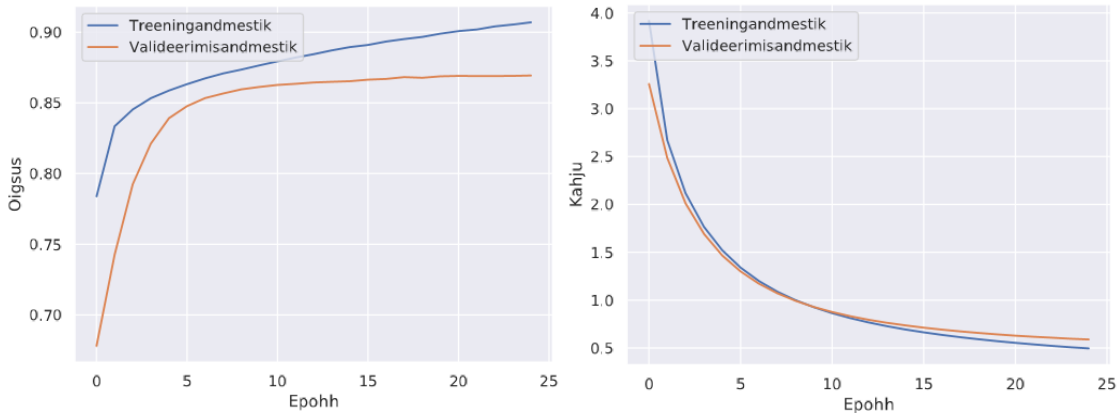
LISA 7. NNA_SMALLEQUALDATA TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 7.7. **NNa_smallequaldata treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NNa_ smallequal data**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 5095 0: 5095	väärtus: usaldusvahemik:	0,88 0,877–0,881	0,55 0,546–0,562	0,4 0,393–0,398
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	1,0 0,996–0,998	0,88 0,876–0,881	0,93 0,932–0,935
Treeniti x korda k-korda	x = 11 k = 10	Viitesuhtes	0,21 0,207–0,213	0,88 0,881–0,887	0,34 0,336–0,342
Ploki suurus	256	Makro keskmine	0,6 0,601–0,604	0,88 0,881–0,883	0,636 0,634–0,639
Epohhe	25	Kaalutud keskmine	0,97 0,97–0,97	0,88 0876–0,88	0,91 0,912–0,914

Tabel 7.2. **NNa_smallequaldata keskmine segadusmaatriks.**

I tüübi viga	0,79	**Mudeli ennustus**	
II tüübi viga	0,0	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	1003,41 88%	131,59 12%
	Viitesuhteta 100%	3791,14 12%	27560,86 88%



Joonis 7.1. **NNa_smallequaldata õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

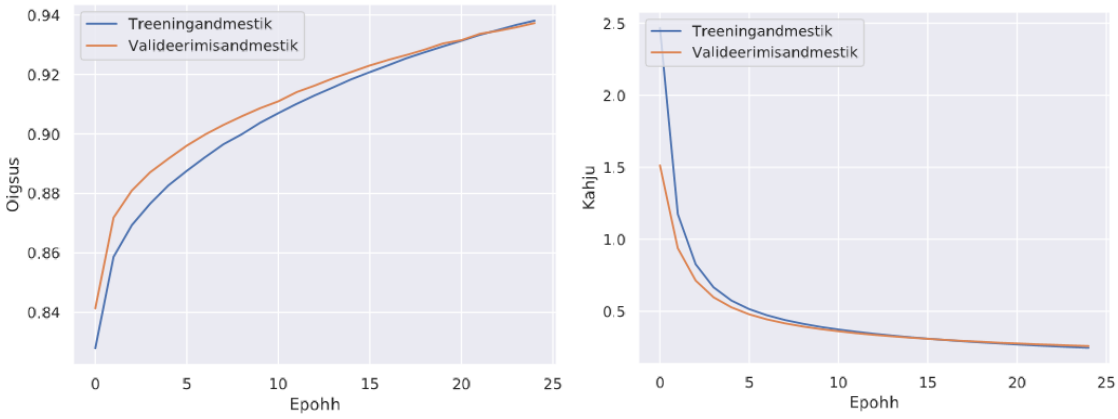
LISA 8. NNA_5EQUALDATA TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 8.8. **NNa_5equaldata treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NNa_ 5equaldata**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 5*5095 0: 25475	väärtus: usaldusvahemik:	0,9 0,9–0,902	0,33 0,328–0,335	0,43 0,431–0,434
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	1,0 0,995–0,997	0,9 0,9–0,903	0,95 0,946–0,948
Treeniti x korda k-korda	x = 15 k = 7	Viitesuhtes	0,25 0,244–0,247	0,88 0,875–0,88	0,38 0,381–0,385
Ploki suurus	256	Makro keskmine	0,62 0,619–0,621	0,89 0,889–0,89	0,66 0,662–0,666
Epohhe	25	Kaalutud keskmine	0,97 0,97–0,97	0,9 0,9–0,902	0,93 0,926–0,928

Tabel 8.2. **NNa_5equaldata keskmine segadusmaatriks.**

I tüübi viga	0,76	**Mudeli ennustus**	
II tüübi viga	0,0	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	995,66 88%	139,34 12%
	Viitesuhteta 100%	3079,83 10%	28272,17 90%



Joonis 8.1. **NNa_5equaldata õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

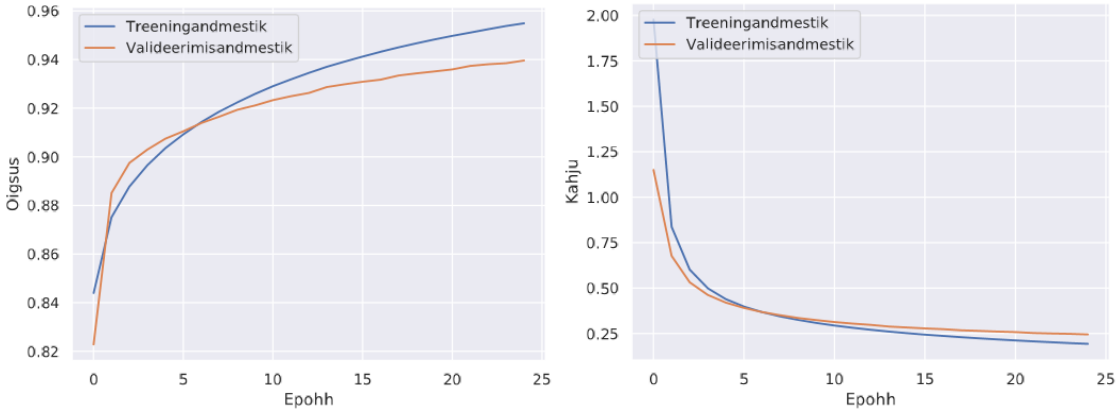
LISA 9. NNA_ADASYN TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 9.9. **NNa_adasyn treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NNa_ adasyn**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 5*8480 0: 42400	väärtus: usaldusvahemik:	0,89 0,889–0,891	0,37 0,368–0,374	0,42 0,416–0,419
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	1,0 0,999–1,0	0,89 0,889–0,891	0,94 0,939–0,941
Treeniti x korda k-korda	x = 15 k = 7	Viitesuhtes	0,23 0,226–0,23	0,89 0,889–0,892	0,36 0,361–0,365
Ploki suurus	256	Makro keskmine	0,61 0,611–0,612	0,89 0,89–0,89	0,65 0,65–0,652
Epohhe	25	Kaalutud keskmine	0,97 0,97–0,97	0,89 0,889–0,891	0,92 0,919–0,92

Tabel 9.2. **NNa_adasyn keskmine segadusmaatriks.**

I tüübi viga	0,77	**Mudeli ennustus**	
II tüübi viga	0,0	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	1010,9 89%	124,1 11%
	Viitesuhteta 100%	3439,77 10%	27912,23 89%



Joonis 9.1. **NNa_adasyn õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

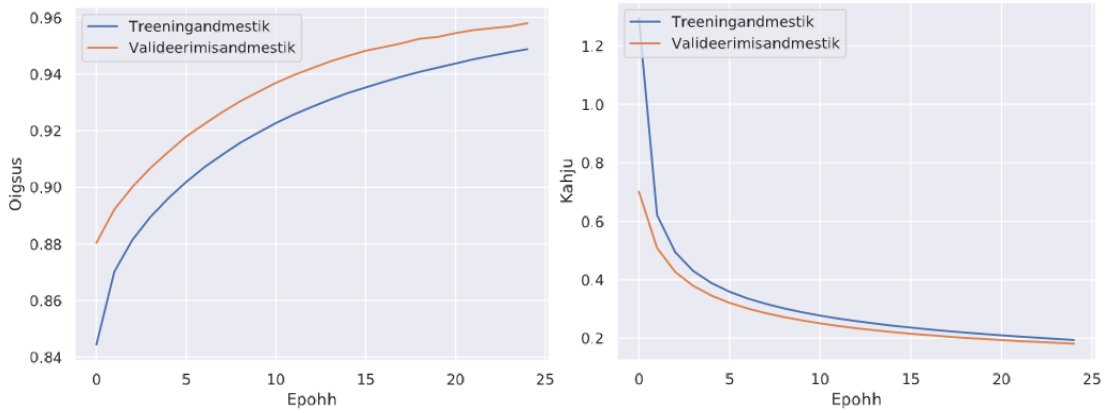
LISA 10. NNA_BATCH64 TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 10.10. **NNa_batch64 treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NNa_ batch64**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,92 0,921–0,923	0,26 0,253–0,258	0,47 0,469–0,472
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	0,99 0,99–0,991	0,92 0,923–0,925	0,96 0,958–0,96
Treeniti x korda k-korda	x = 15 k = 7	Viitesuhtes	0,29 0,287–0,292	0,85 0,852–0,857	0,43 0,431–0,436
Ploki suurus	64	Makro keskmine	0,64 0,641–0,644	0,89 0,889–0,89	0,7 0,694–0,697
Epohhe	25	Kaalutud keskmine	0,97 0,97–0,97	0,92 0,921–0,922	0,94 0,939–0,941

Tabel 10.2. **NNa_batch64 keskmine segadusmaatriks.**

I tüübi viga	0,71	**Mudeli ennustus**	
II tüübi viga	0,01	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	969,32 85%	165,68 15%
	Viitesuhteta 100%	2375.94 8%	28976,06 92%



Joonis 10.1. **NNa_batch64 õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

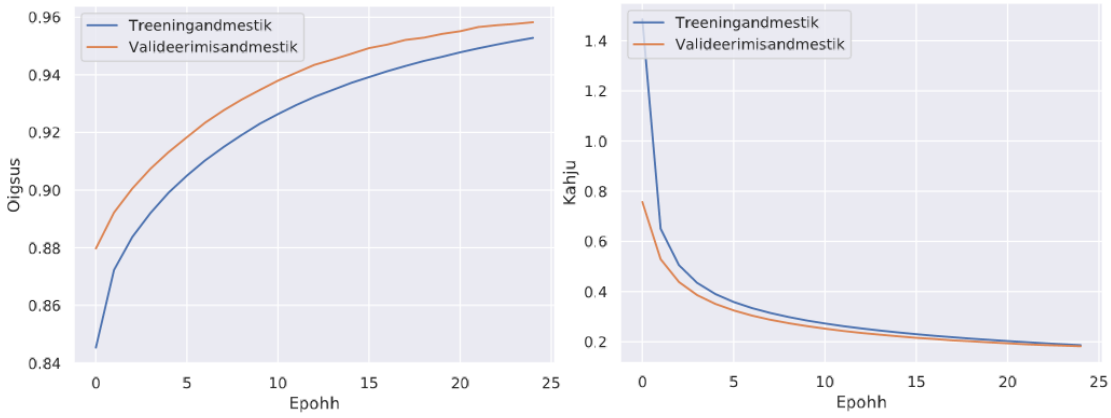
LISA 11. NNA_BATCH128 TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 11.11. **NNa_batch128 treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NNa_ batch128**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,92 0,921–0,923	0,26 0,253–0,258	0,47 0,469–0,472
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	0,99 0,99–0,99	0,92 0,923–0,925	0,96 0,959–0,96
Treeniti x korda k-korda	x = 15 k = 7	Viitesuhtes	0,29 0,288–0,294	0,85 0,849–0,851	0,43 0,43–0,435
Ploki suurus	128	Makro keskmine	0,64 0,641–0,644	0,89 0,887–0,888	0,7 0,694–0,697
Epohhe	25	Kaalutud keskmine	0,97 0,97–0,97	0,92 0,921–0,923	0,94 0,939–0,94

Tabel 11.2. **NNa_batch128 keskmine segadusmaatriks.**

I tüübi viga	0,71	**Mudeli ennustus**	
II tüübi viga	0,01	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	965,2 85%	169,8 15%
	Viitesuhteta 100%	2366,11 8%	28985,89 92%



Joonis 11.1. **NNa_batch128 õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

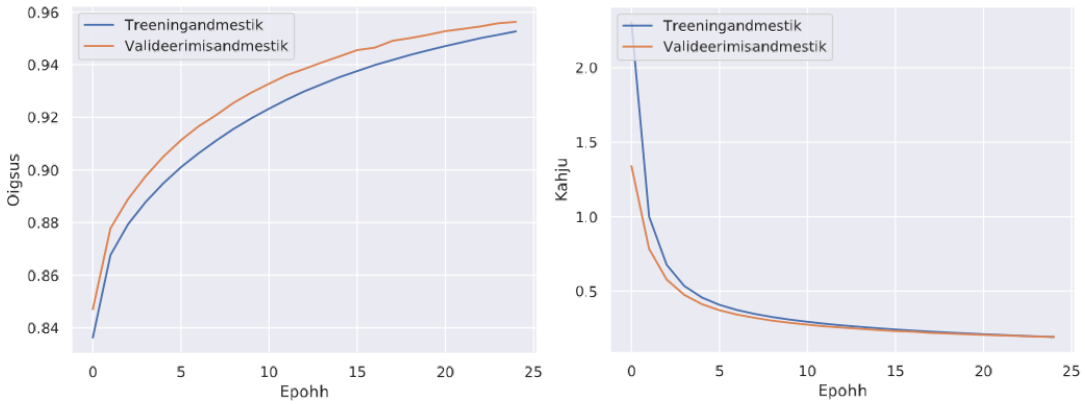
LISA 12. NNA_BATCH512 TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 12.12. **NNa_batch512 treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NNa_ batch512**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,92 0,919–0,921	0,27 0,266–0,272	0,47 0,463–0,468
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	0,99 0,99–0,99	0,92 0,92–0,92	0,96 0,957–0,959
Treeniti x korda k-korda	x = 15 k = 7	Viitesuhtes	0,29 0,283–0,288	0,85 0,851–0,856	0,43 0,424–0,43
Ploki suurus	512	Makro keskmine	0,64 0,639–0,641	0,89 0,888–0,889	0,69 0,69–0,694
Epohhe	25	Kaalutud keskmine	0,97 0,97–0,97	0,92 0,919–0,921	0,94 0,938–0,94

Tabel 12.2. **NNa_batch512 keskmine segadusmaatriks.**

I tüübi viga	0,72	**Mudeli ennustus**	
II tüübi viga	0,01	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	966,05 85%	165,95 15%
	Viitesuhteta 100%	2437,89 8%	28914,11 92%



Joonis 12.1. **NNa_batch512 õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

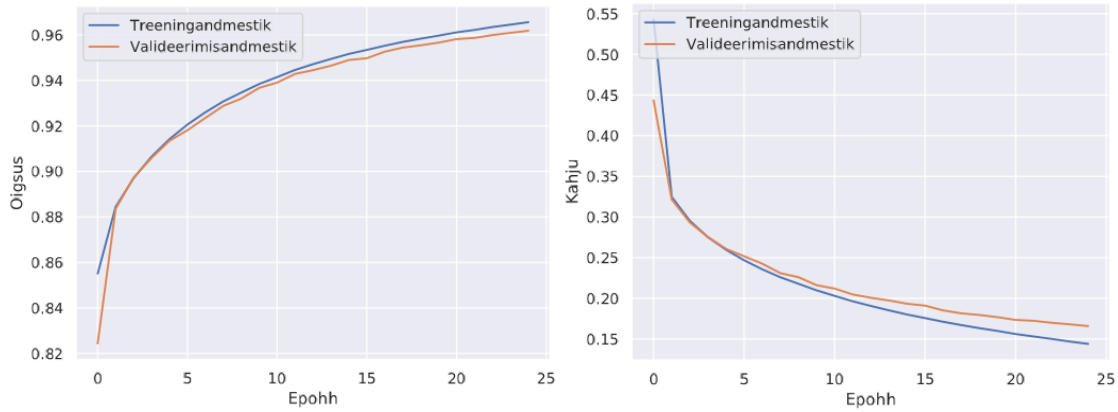
LISA 13. NNA_LR01 TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 13.13. **NNa_lr01 treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NNa_ lr01**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,93 0,931–0,934	0,25 0,246–0,255	0,48 0,474–0,482
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	0,99 0,99–0,99	0,94 0,934–0,939	0,97 0,964–0,966
Treeniti x korda k-korda	x = 8 k = 7	Viitesuhtes	0,32 0,312–0,323	0,8 0,797–0,807	0,46 0,45–0,46
Ploki suurus	256	Makro keskmine	0,65 0,652–0,658	0,87 0,867–0,871	0,71 0,706–0,712
Epohhe	25	Kaalutud keskmine	0,97 0,97–0,97	0,93 0,93–0,935	0,95 0,945–0,947

Tabel 13.2. **NNa_lr01 keskmine segadusmaatriks.**

I tüübi viga	0,68	**Mudeli ennustus**	
II tüübi viga	0,01	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	910,04 80%	224,96 20%
	Viitesuhteta 100%	1972,43 6%	29379,52 94%



Joonis 13.1. **NNa_lr01 õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

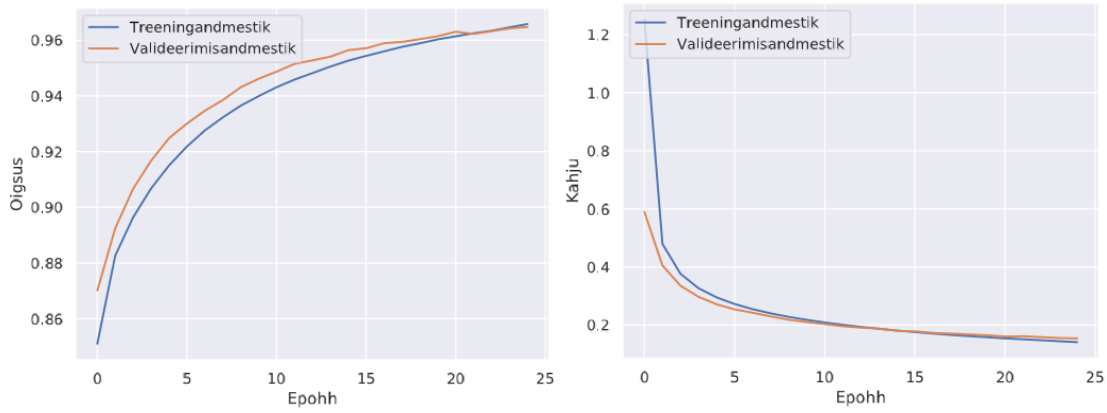
LISA 14. NNA_LR003 TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 14.14. **NNa_lr003 treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NNa_ lr003**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,93 0,93–0,932	0,24 0,238–0,245	0,48 0,48–0,486
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	0,99 0,99–0,99	0,94 0,933–0,936	0,96 0,962–0,964
Treeniti x korda k-korda	x = 8 k = 7	Viitesuhtes	0,32 0,312–0,319	0,82 0,817–0,827	0,46 0,451–0,458
Ploki suurus	256	Makro keskmine	0,65 0,652–0,656	0,88 0,876–0,881	0,71 0,706–0,712
Epohhe	25	Kaalutud keskmine	0,97 0,97–0,97	0,931 0,929–0,933	0,95 0,944–0,946

Tabel 14.2. **NNa_lr003 keskmine segadusmaatriks.**

I tüübi viga	0,69	**Mudeli ennustus**	
II tüübi viga	0,01	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	933,11 82%	201,89 18%
	Viitesuhteta 100%	2039,23 7%	29312,77 93%



Joonis 14.1. **NNa_lr003 õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

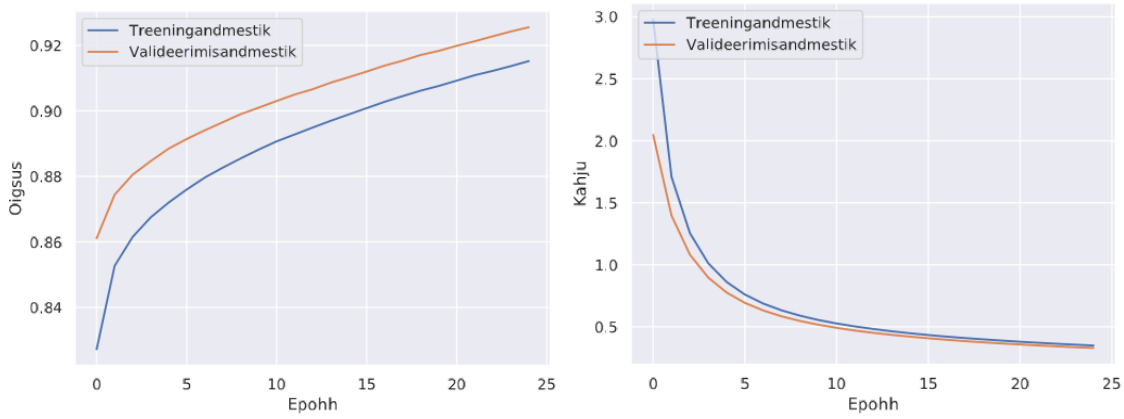
LISA 15. NNA_LR001 TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 15.15. **NNa_lr001 treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NNa_ lr001**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,89 0,89–0,892	0,39 0,387–0,394	0,42 0,419–0,421
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	1,0 1,0–1,0	0,89 0,94–0,941	0,94 0,94–0,941
Treeniti x korda k-korda	x = 8 k = 7	Viitesuhtes	0,23 0,228–0,331	0,89 0,892–0,896	0,37 0,363–0,366
Ploki suurus	256	Makro keskmine	0,61 0,611–0,612	0,89 0,89–0,891	0,65 0,651–0,653
Epohhe	25	Kaalutud keskmine	0,97 0,97–0,97	0,89 0,89–0,89	0,92 0,92–0,921

Tabel 15.2. **NNa_lr001 keskmine segadusmaatriks.**

I tüübi viga	0,77	**Mudeli ennustus**	
II tüübi viga	0,0	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	1014,91 89%	120,09 11%
	Viitesuhteta 100%	3420,77 11%	27931,23 89%



Joonis 15.1. **NNa_lr001 õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

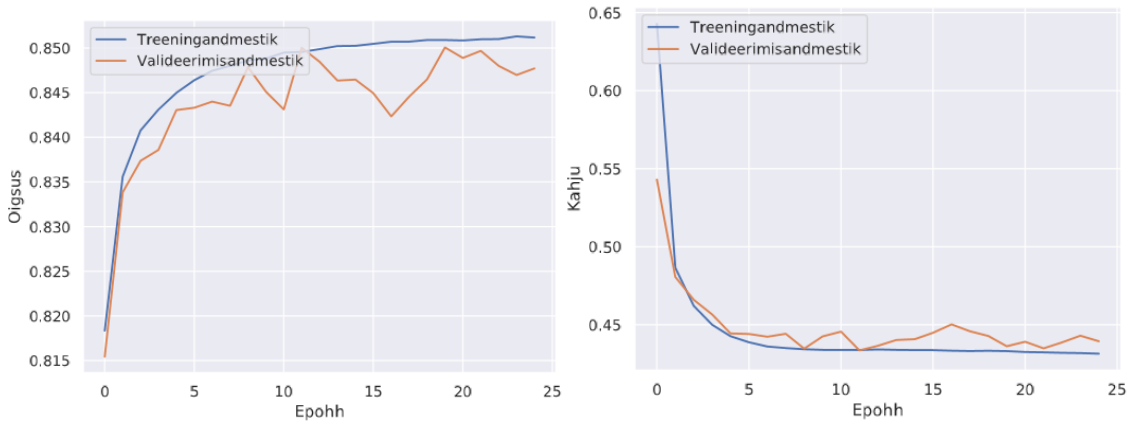
LISA 16. NN1_LR01 TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 16.16. **NN1_lr01 treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NN1_ lr01**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,84 0,822–0,851	0,43 0,402–0,463	0,35 0,336–0,358
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	1,0 0,995–0,997	0,83 0,819–0,85	0,91 0,897–0,916
Treeniti x korda k-korda	x = 8 k = 7	Viitesuhtes	0,18 0,166–0,185	0,88 0,871–0,969	0,29 0,275–0,302
Ploki suurus	256	Makro keskmine	0,59 0,58–0,59	0,86 0,856–0,863	0,6 0,586–0,609
Epohhe	25	Kaalutud keskmine	0,97 0,969–0,97	0,84 0,822–0,85	0,89 0,876–0,895

Tabel 16.2. **NN1_lr01 keskmine segadusmaatriks.**

I tüübi viga	0,84	**Mudeli ennustus**	
II tüübi viga	0,01	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	1002,04 88%	132,96 12%
	Viitesuhteta 100%	5186,39 17%	26165,61 83%



Joonis 16.1. **NN1_lr01 õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

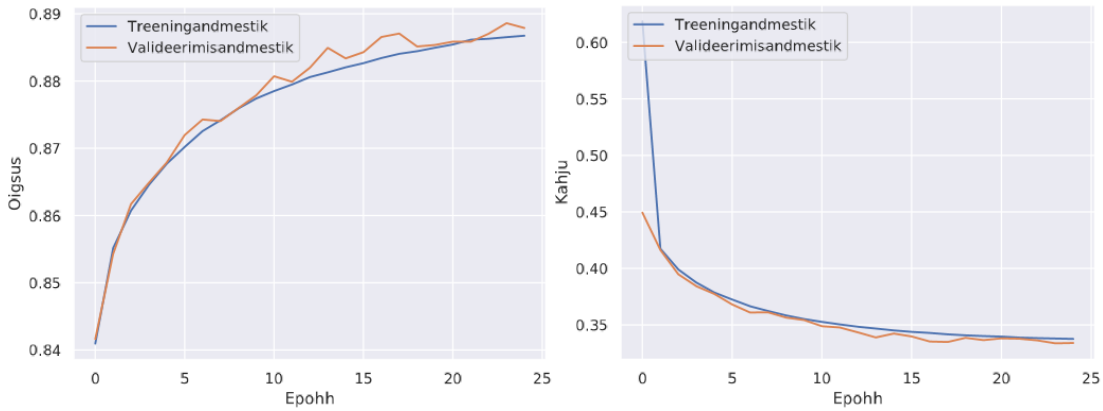
LISA 17. NN1_LR003 TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 17.17. **NN1_lr003 treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NN1_ lr003**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,87 0,863–0,88	0,34 0,32–0,353	0,39 0,378–0,398
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	1,0 0,996–0,998	0,87 0,863–0,88	0,93 0,924–0,933
Treeniti x korda k-korda	x = 8 k = 7	Viitesuhtes	0,21 0,196–0,217	0,88 0,875–0,893	0,33 0,319–0,345
Ploki suurus	256	Makro keskmine	0,6 0,595–0,606	0,88 0,875–0,88	0,63 0,621–0,639
Epohhe	25	Kaalutud keskmine	0,97 0,97–0,97	0,87 0,864–0,88	0,91 0,903–0,913

Tabel 317.2. **NN1_lr003 keskmine segadusmaatriks.**

I tüübi viga	0,8	**Mudeli ennustus**	
II tüübi viga	0,0	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	1002,34 88%	132,66 12%
	Viitesuhteta 100%	4044,61 13%	27307,39 87%



Joonis 17.1. **NN1_lr003 õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

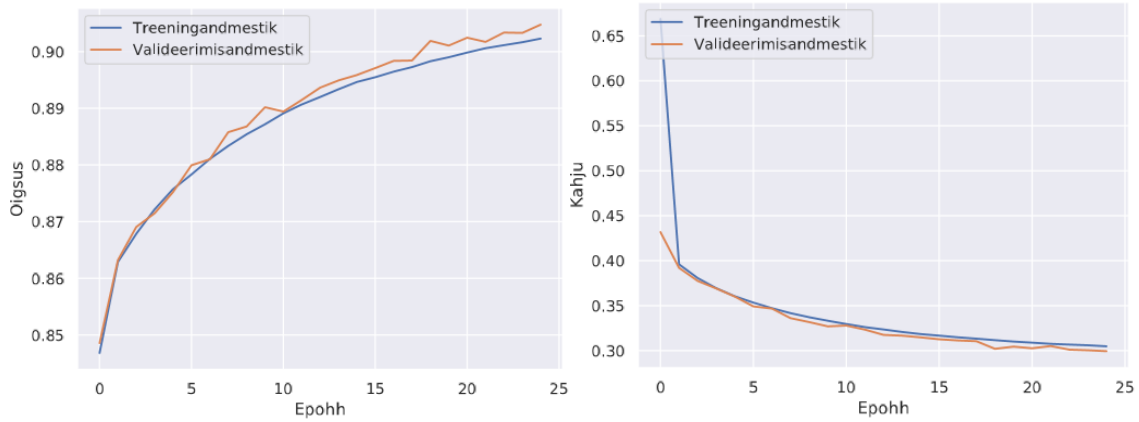
LISA 18. NN1_LR002 TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 18.18. **NN1_lr002 treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NN1_ lr002**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,88 0,874–0,888	0,32 0,31–0,337	0,4 0,393–0,408
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	1,0 0.995–0,998	0,88 0,873–0,889	0,94 0,93–0,939
Treeniti x korda k-korda	x = 8 k = 7	Viitesuhtes	0,21 0,208–0,224	0,88 0,875–0,89	0,35 0,335–0,356
Ploki suurus	256	Makro keskmine	0,61 0,601–0,61	0,88 0,88–0,884	0,64 0,633–0,647
Epohhe	25	Kaalutud keskmine	0,97 0,97–0,97	0,88 0,874–0,888	0,91 0,908–0,918

Tabel 18.2. **NN1_lr002 keskmine segadusmaatriks.**

I tüübi viga	0,79	**Mudeli ennustus**	
II tüübi viga	0,0	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	1001,86 88%	133,14 12%
	Viitesuhteta 100%	3744,7 12%	27607,3 88%



Joonis 18.1. **NN1_lr002 õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

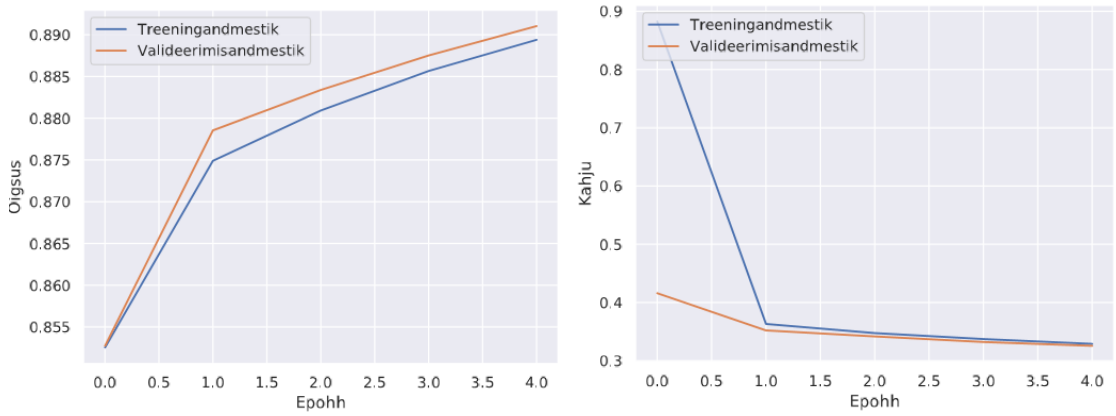
LISA 19. NN1_EPOCH5 TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 19.19. **NN1_epoch5 treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NN1_ epoch5**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,87 0,863–0,876	0,35 0,337–0,364	0,39 0,38–0,393
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	1,0 0.996–0,998	0,87 0,862–0,876	0,93 0,924–0,932
Treeniti x korda k-korda	x = 15 k = 7	Viitesuhtes	0,2 0,195–0,208	0,89 0,884–0,897	0,33 0,319–0,337
Ploki suurus	256	Makro keskmine	0,6 0,595–0,602	0,88 0,879–0,881	0,63 0,621–0,635
Epohhe	5	Kaalutud keskmine	0,97 0,97–0,97	0,87 0,863–0,876	0,91 0,902–0,911

Tabel 19.2. **NN1_epoch5 keskmine segadusmaatriks.**

I tüübi viga	0,8	**Mudeli ennustus**	
II tüübi viga	0,0	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	1010,37 89%	124,63 11%
	Viitesuhteta 100%	4104,63 13%	27247,37 87%



Joonis 19.1. **NN1_epoch5 õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

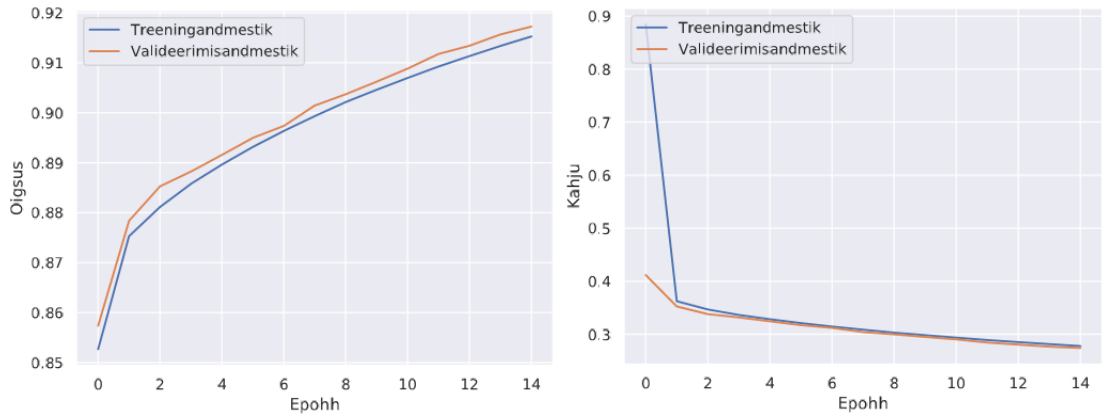
LISA 20. NN1_EPOCH15 TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 20.20. **NN1_epoch15 treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NN1_ epoch15**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,9 0,893–0,899	0,3 0,297–0,312	0,42 0,416–0,425
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	0,99 0.993–0,994	0,9 0,893–0,901	0,94 0,941–0,945
Treeniti x korda k-korda	x = 15 k = 7	Viitesuhtes	0,24 0,232–0,243	0,87 0,861–0,872	0,37 0,366–0,379
Ploki suurus	256	Makro keskmine	0,62 0,614–0,619	0,88 0,881–0,884	0,66 0,653–0,662
Epohhe	15	Kaalutud keskmine	0,97 0,97–0,97	0,9 0,892–0,9	0,92 0,922–0,926

Tabel 20.2. **NN1_epoch15 keskmine segadusmaatriks.**

I tüübi viga	0,77	**Mudeli ennustus**	
II tüübi viga	0,01	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	984,09 87%	150,91 13%
	Viitesuhteta 100%	3222,77 10%	28129,23 90%



Joonis 20.1. **NN1_epoch15 õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

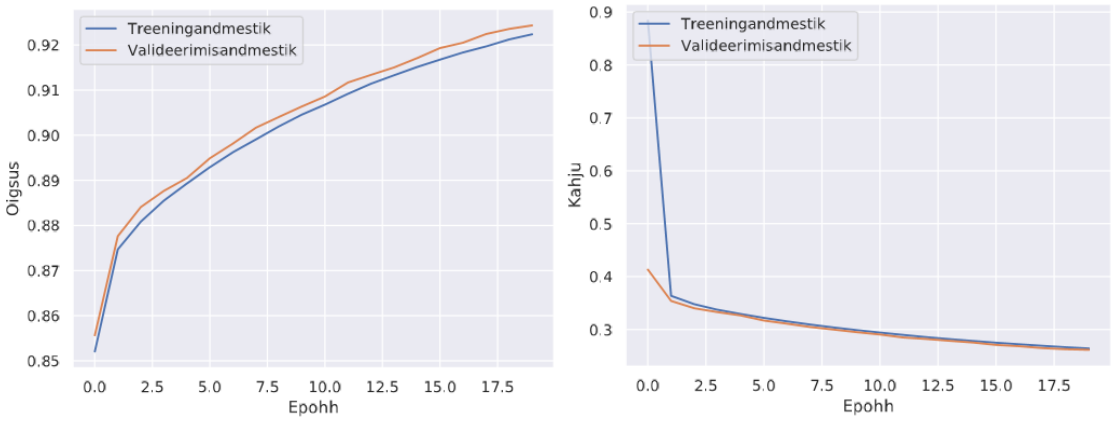
LISA 21. NN1_EPOCH20 TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 21.21. **NN1_epoch20 treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NN1_ epoch20**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,9 0,898–0,906	0,3 0,29–0,308	0,43 0,424–0,435
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	0,99 0.992–0,995	0,9 0,899–0,908	0,95 0,944–0,949
Treeniti x korda k-korda	x = 15 k = 7	Viitesuhtes	0,25 0,241–0,255	0,86 0,853–0,867	0,38 0,376–0,391
Ploki suurus	256	Makro keskmine	0,62 0,617–0,624	0,88 0,881–0,884	0,67 0,66–0,67
Epohhe	20	Kaalutud keskmine	0,97 0,97–0,97	0,9 0,898–0,905	0,93 0,924–0,929

Tabel 21.2. **NN1_epoch20 keskmine segadusmaatriks.**

I tüübi viga	0,76	**Mudeli ennustus**	
II tüübi viga	0,01	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	976,75 86%	158,25 14%
	Viitesuhteta 100%	3033,5 10%	28318,5 90%



Joonis 21.1. **NN1_epoch20 õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

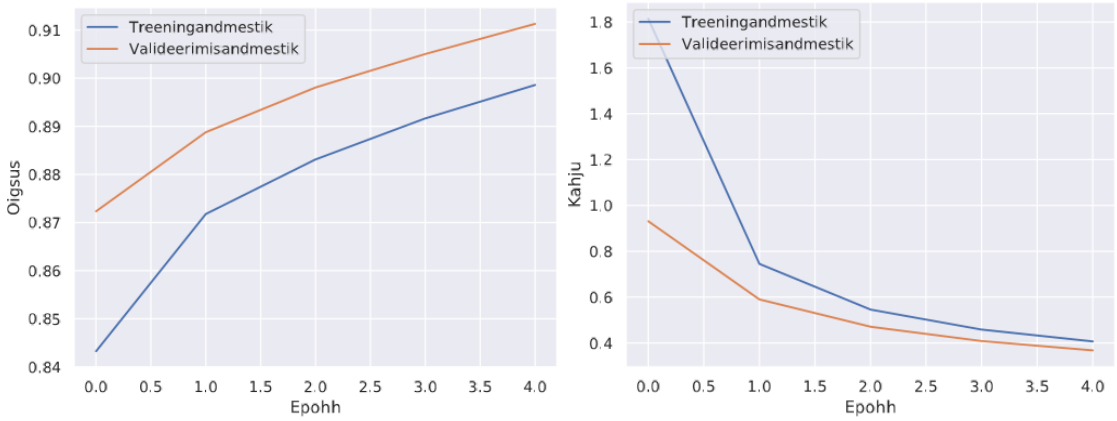
LISA 22. NNA_EPOCH5 TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 22.22. **NNa_epoch5 treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NNa_ epoch5**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,89 0,891–0,893	0,39 0,391–0,397	0,42 0,419–0,422
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	1,0 0.999–1,0	0,89 0,891–0,894	0,94 0,94–0,941
Treeniti x korda k-korda	x = 15 k = 7	Viitesuhtes	0,23 0,229–0,233	0,89 0,887–0,89	0,37 0,364–0,368
Ploki suurus	256	Makro keskmine	0,61 0,612–0,615	0,89 0,889–0,89	0,65 0,652–0,655
Epohhe	5	Kaalutud keskmine	0,97 0,97–0,97	0,89 0,891–0,894	0,92 0,92–0,921

Tabel 22.2. **NNa_epoch5 keskmine segadusmaatriks.**

I tüübi viga	0,77	**Mudeli ennustus**	
II tüübi viga	0,0	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	1008,07 89%	126,93 11%
	Viitesuhteta 100%	3371,79 11%	27980,21 89%



Joonis 22.1. **NNa_epoch5 õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

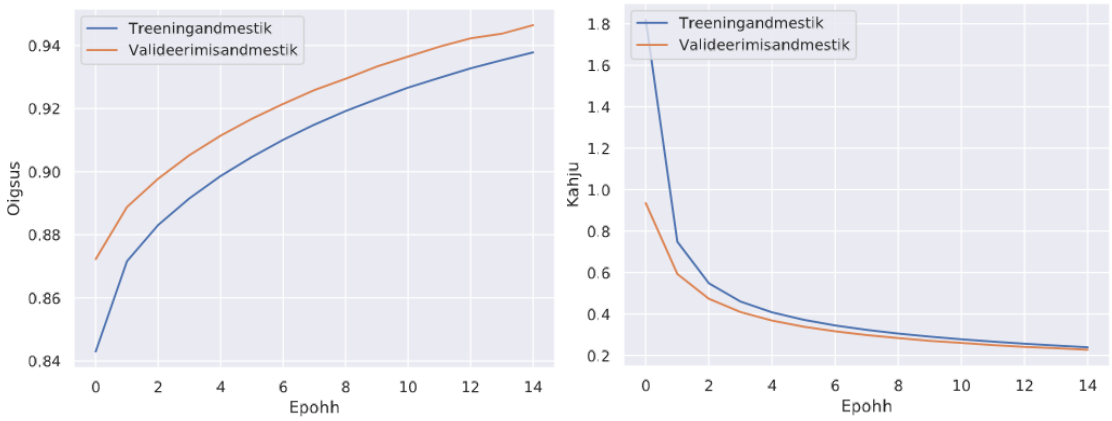
LISA 23. NNA_EPOCH15 TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 23.23. **NNa_epoch15 treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NN1_ epoch15**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,91 0,908–0,911	0,3 0,291–0,299	0,45 0,445–0,451
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	0,99 0.992–0,994	0,91 0,909–0,913	0,95 0,95–0,952
Treeniti x korda k-korda	x = 15 k = 7	Viitesuhtes	0,26 0,26–0,266	0,87 0,868–0,871	0,4 0,4–0,407
Ploki suurus	256	Makro keskmine	0,63 0,627–0,631	0,89 0,889–0,891	0,68 0,675–0,68
Epohhe	15	Kaalutud keskmine	0,97 0,97–0,97	0,91 0,908–0,912	0,93 0,931–0,933

Tabel 23.2. **NNa_epoch15 keskmine segadusmaatriks.**

I tüübi viga	0,74	**Mudeli ennustus**	
II tüübi viga	0,01	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	986,41 87%	148,59 13%
	Viitesuhteta 100%	2780,34 9%	28571,66 91%



Joonis 23.1. **NNa_epoch15 õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

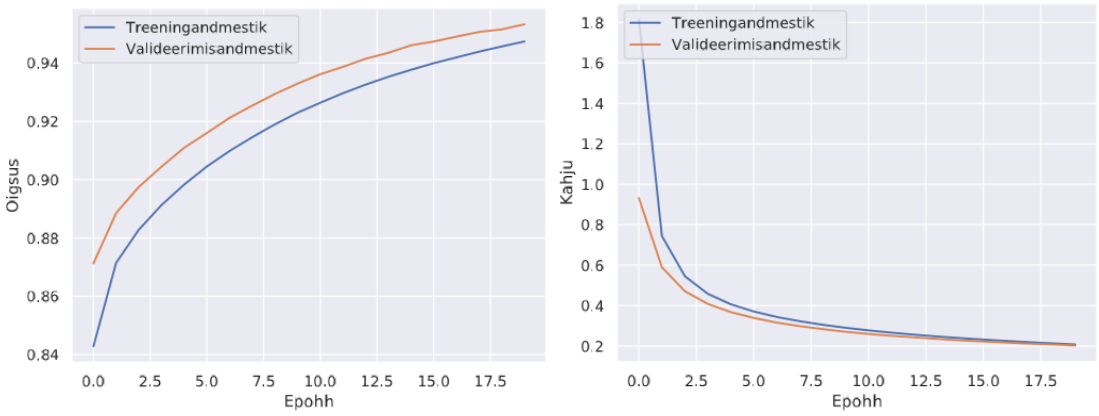
LISA 24. NNA_EPOCH20 TULEMUSED TASAKAALUTA TESTANDMESTIKUL

Tabel 24.24. **NNa_epoch20 treenimisandmed ja keskmised tulemused.**

Treenimis-andmed	**NNa_ epoch20**	**Testandmestiku tulemused**	**Õigsus**	**Kahju**	**MCC**
Test- ja treening-andmestik	1: 10*5095 0: 51010	väärtus: usaldusvahemik:	0,92 0,915–0,918	0,28 0,274–0,28	0,46 0,457–0,462
			Täpsus	**Saagis**	**F1**
Valideerimis-andmestik	1: 1135 0: 31352	Viitesuhteta	0,99 0.991–0,992	0,92 0,918–0,92	0,96 0,954–0,956
Treeniti x korda k-korda	x = 15 k = 7	Viitesuhtes	0,28 0,274–0,28	0,86 0,858–0,862	0,42 0,415–0,422
Ploki suurus	256	Makro keskmine	0,64 0,634–0,638	0,89 0,888–0,89	0,69 0,685–0,689
Epohhe	20	Kaalutud keskmine	0,97 0,97–0,97	0,92 0,915–0,918	0,94 0,935–0,938

Tabel 24.2. **NNa_epoch20 keskmine segadusmaatriks.**

I tüübi viga	0,72	**Mudeli ennustus**	
II tüübi viga	0,01	Viitesuhtes	Viitesuhteta
Tegelikult	Viitesuhtes 100%	975,24 86%	159,76 14%
	Viitesuhteta 100%	2556,4 8%	28795,6 92%



Joonis 24.1. **NNa_epoch20 õigsuse ja kahju muutused treening- ja valideerimisandmestikul epohhide lõikes.**

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Linda Freienthal (sünnikuupäev: 27.02.1996),

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Pronominaalsete viitesuhete automaatne lahendamine eesti keeles närvivõrkude abil“, mille juhendaja on Kadri Muischnek,

1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, 24.05.2020