

Opinnäytetyö (AMK)

Tieto- ja viestintäteknikka

2022

Kaspar Kaasikoja

Silmätautipotilaiden tilan ennustaminen long short-term memory-menetelmällä



Opinnäytetyö (AMK) | Tiivistelmä

Turun ammattikorkeakoulu

Tieto- ja viestintätekniikka

2022 | 48 sivua

Kaspar Kaasikoja

Silmätautipotilaiden tilan ennustaminen long short-term memory-menetelmällä

Silmänpohjan ikärappeuma on maailmanlaajuisesti yleisin näkövammaisuutta aiheuttava sairaus, jota esiintyy hoidettavana kosteana ja hoitamattomana kuivana muotona. Kosteaa etenemistä hidastavat pistoshoidot ovat lisääntyneet tehokkuutensa vuoksi. Hoito- ja seurantakäyntien potilastiedot antavat pohjan koneoppimisen hyödyntämiseen. Näiden hyödyntäminen on lisääntynyt, mutta niiden haasteena on ollut huomioida pitkät aikavälit, joita voi olla potilastiedoissa paljon.

Tämän työn tavoitteena oli kokeilla pitkiin aikaväleihin erikoistunutta long short-term memory-neuroverkkomenetelmän soveltuvuutta synteettiseen kostean silmänpohjan ikärappeumapotilasdataan. Tällä pyrittiin ennustamaan potilaiden tilan muuttumista parempaan tai huonompaan suuntaan seuraavalle käyntikerralle.

Menetelmällä saavutettiin erinomainen yli 90 %:n tarkkuus, testatessa sitä isolla synteettisellä potilasaineistolla, joka jaettiin suhteella 67:33 treenidataan ja testidataan. Tämän perusteella menetelmää voitaisiin hyödyntää myös aitoon potilasdataan.

Tutkimusta voisi jatkaa ottamalla pistosvaihtoehdot mukaan, mikä auttaisi lääkkeen valitsemisessa potilaalle. Toinen jatkokehitys voisi olla hoitojen kustannuskehityksen ennustaminen.

Asiasanat:

silmänpohjan nesteinen ikärappeuma, long short-term memory, valvottu koneoppiminen, neuroverkot

Bachelor's Thesis | Abstract

Turku University of Applied Sciences

Bachelor of Engineering, Information and Communications Technology

2022 | 48 pages

Kaspar Kaasikoja

Predicting the condition of age-related macular degeneration patients with long short-term memory

Age-related macular degeneration (AMD) is the most common cause of visual impairment worldwide, occurring in treatable wet form (wAMD) and untreatable dry form (dry AMD). wAMD injection treatments has increased due to their effectiveness, which in turn increases health care costs. Treatment and follow-up visit patient data provides a basis for utilizing machine learning (ML). ML usage has increased but the challenge has been to consider long-term periods that may be in patient data.

The purpose of this thesis was to test the applicability of the long short-term memory neural network method, whose speciality is in long-term dependencies, based on synthetic wAMD patient data. The aim was to predict the patients' condition for the next visit.

The method was tested on the large synthetic patient data, splitted into 67:33 test and train data. The result was that the method was significantly accurate, over 90 %. Based on this, this method could be utilized on real-world data.

Injectables could be included in further development so that the method could propose injectables for patients. Another development could be predicting treatment cost development.

Keywords:

neovascular age-related macular degeneration, long short-term memory, supervised machine learning, neural networks

Sisältö

Käytetyt lyhenteet ja sanasto	5
1 Johdanto	9
2 Silmämpohjan ikärappeuma	11
2.1 Oireet	11
2.2 Tärkeimmät riskitekijät	12
2.3 Tutkiminen	12
2.4 Hoito ja seuranta	13
3 Neuroverkot	15
3.1 Neuron	16
3.2 Myötäkytketty neuroverkko	19
3.3 Takaisinkytketty neuroverkko	19
3.4 Katoavan gradientin ongelma	21
4 LSTM-neuroverkko	23
4.1 LSTM-muistisolu	23
4.2 LSTM:n variaatiot	27
5 Ennustuskokeilu ja tulokset	28
5.1 Ohjelmointityövälineet ja -kirjastot	28
5.2 Vanilla LSTM-mallin rakentaminen	29
5.2.1 Käytetty tappiofunktio	30
5.2.2 Käytetty optimoija	30
5.3 Vanilla LSTM-mallin oppiminen	31
5.4 Vanilla LSTM-mallin tulokset	33
6 Lopuksi	34
Lähteet	36

Käytetyt lyhenteet ja sanasto

\tanh	hyperbolinen tangenttifunktio (hyperbolic tangent)
σ	sigmoidifunktio (sigmoid)
t	ajan hetki (time)
w	neuronin painokerroin (weight)
x	koneoppimisessa ominaisuudet, jotka vaikuttavat kohteeseen y (x , features)
y	koneoppimisessa kohde, joka saadaan ominaisuuksista x (y , target)
Adam	koneoppimismallin tappion pienentämiseen käytettävä mukautuva hetken arviointioptimointialgoritmi (Adaptive Moment Estimation)
AI	tekoäly (artificial intelligence)
aika-askel	ilmaisee mennyttä, nykyistä ja tulevaa ajanhetkeä (time step)
aktivaatio	tiedon tärkeydestä riippuva aktivaatiofunktiolla tapahtuva neuronin aktivoituminen, joka tuottaa sen ulostulon (activation)
alisovitus	koneoppimismalli ei ymmärrä syötteen ja ulostulon muuttujien yhteyttä tarkasti (underfitting)
AMD	silmänpohjan ikärappeuma (age-related macular degeneration)
ANN	keinoverkko (artificial neural network)
BRNN	bidirektionaalinen neuroverkko (bidirectional neural network)
CNN	konvoluutioinen neuroverkko (convolutional neural network)

DL	syväoppiminen (deep learning)
epoch	koneoppimismallin opetusdatan näkemiskierrokset oppimisen aikana
FAG	fluoreseiiniväriainetutkimus (fluorescent angiography)
Fimea	lääkealan turvallisuus- ja kehittämiskeskus (Finnish Medicines Agency)
fixed	suunnitellusti säännöllisin väliajoin annosteltava hoitoprotokolla
FNN	myötäkytketty neuroverkko (feedforward neural network)
gradientti	ilmaisee funktion sijainnin, suurimman muutosnopeuden ja suurimman muutoksen suunnan
gradienttilasku	koneoppimismallin tappiofunktion pienentämiseen käytettävä yleinen optimointialgoritmi (gradient descent)
GRU	takaisinkytketyn neuroverkon porttimekanismi (gated recurrent unit)
hyperparametri	koneoppimismallin oppimistarkkuuteen vaikuttavat muokattavissa olevat opittavat parametrit
hämmennysmatriisi	kuvaava koneoppimismallin oikein ja väärin luokiteltuja tai ennustettuja arvoja (confusion matrix)
IBM	International Business Machines Corporation
ICG	indosyainivihreäväriainetutkimus (indocyanine green)
Jupyter	web-pohjainen muistiotyylinen interaktiivinen ohjelmointiympäristö
kerros	neuronien järjestys omissa kerroksissa neuroverkkoarkkitehtuurissa, kuten sisääntulo-, piilo- tai ulostulokerros (layer)
LSTM	pitkä lyhytaikainen muisti (long short-term memory)

MIT	Massachusetts Institute of Technology
ML	koneoppiminen (machine learning)
MSE	koneoppimismallin tappiofunktiona käytettävä keskimääräinen neliövirhe (mean squared error)
neuroni	neuroverkon kerroksissa olevat tiedon kulkua säätelevät laskentayksiköt (neuron, node)
NN	neuroverkko (neural network)
OCT	verkkokalvon valokerroskuvaus (optical coherence tomography)
PDT	valoaktivaatiohoito (photodynamic therapy)
piilokerros	sisääntulo- ja ulostulokerroksien välissä oleva kerros neuroverkkoarkkitehtuurissa (hidden layer)
PRN	tarvittaessa annosteltava hoitoprotokolla (pro re nata)
Python 3	funktionaalisen olio-ohjelmointikielen versio 3
RNN	takaisinkytketty neuroverkko (recurrent neural network)
syväoppiminen	useammalla piilokerroksella varustettu neuroverkko (deep learning)
TAE	tilanteen mukaan vähitellen pidentyvin väliajoin annosteltava hoitoprotokolla (treat and extend)
tappio	koneoppimismallin tappio prosentteina (loss)
tappiofunktio	laskee koneoppimismallin tappion prosentteina (loss function)
tarkkuus	koneoppimismallin tarkkuus prosentteina (accuracy)
Vanilla LSTM	yksinkertainen LSTM-neuroverkkomalli
vastavirta-algoritmi	laskee tappiofunktion gradientteja edeten viimeisestä piilokerroksesta ensimmäiseen kerrokseen neuroverkon oppiessa (backpropagation)

VEGF	verisuonikasvutekijä (vascular endothelial growth factor)
vektori	suuruuden ja suunnan sisältävä geometrinen malli, jos sisältää myös ulottuvuuden, tulee siitä matriisi
wAMD	nestainen silmänpohjan ikärappeuma (wet age-related macular degeneration)
ylisovitus	koneoppimismalli yleistyy täsmälleen opetusdataan ja sitä ei voida käyttää toiseen dataan (overfitting)

1 Johdanto

Silmänpohjan ikärappeuma kattaa 7–8 % maailman sokeista ja on yleinen kehittyneissä maissa varsinkin yli 60-vuotiailla. Väestön kasvun ja sen eksponentiaalisen ikääntymisen vuoksi sen arvioidaan vuoteen 2050 mennessä olevan merkittävä näköä uhkaava sairaus vanhuksilla maailmanlaajuisesti. Tästä kärsiviä ihmisiä on arvioitu olevan noin 196 miljoonaa ja vuoteen 2040 on ennustettu nousevan lähes 300 miljoonaan. (Wong ym. 2014; BrightFocus Foundation 2021; Zhang ym. 2022.) Tämä aiheuttaa terveydenhuollolle huomattavia lisäkustannuksia, koska taudin aiheuttaman maailmanlaajuisen kustannuksen arvioidaan olevan yli 340 miljardia dollaria, joista yli 250 miljardia dollaria ovat suoria terveydenhoitokustannuksia (Rein ym. 2009; BrightFocus Foundation 2021).

Euroopassa arvioidaan 67 miljoonan ihmisen sairastavan tautia ja ennustetaan sairastavien määrän nousevan 10 miljoonalla vuoteen 2050 mennessä (Li ym. 2020). Suomessa sairauden näkövammaisdiagnoosit ovat olleet nousussa ikääntyneiden lisääntymisen myötä, joista yli 60 % on silmänpohjan ikärappeuman diagnooseja (Ojamo & Tolkkinen 2020, 20–29, 40–43). Fimean (2020) mukaan se on kehitysmaiden yleisin näkövammaisuutta aiheuttava tauti. Tapauksista n. 10–20 % sairastaa taudin kosteaa muotoa (Härkönen & Kiviniemi 2020, 4).

Suomessa on annettu yli 150 000 lasiaisinjektiota kostean muodon hoitamiseen vuosina 2010–2016 (Loukovaara & Haukka 2021). Hoitotulokset ovat parantuneet lasiaisinjektioiden myötä, mutta kasvattaneet kuormitusta silmäklinikoilla, sillä injisointimäärät ovat satakertaistuneet 13 vuodessa (Jørstad ym. 2020; Kivinen & Kaarniranta 2021).

Erilaisten koneoppimismenetelmien käyttö ja niiden soveltaminen lukemattomiin arkisiin käyttötarkoituksiin ovat lisääntyneet viime vuosina (Sarker 2021). Tämän on mahdollistanut grafiikkasuorittimien vuosittainen suorituskyvyn kasvu, joka taas mahdollistaa keinoneuroverkkojen rinnakkaisten painokerrointen laskutoimituksien suorittamisen (Baji 2018). Ihmiset muistavat

tärkeitä tapahtumia hyvinkin pitkiltä ajoilta, mutta tavallisille koneoppimismenetelmille tämä on haaste. Sepp Hochreiterin ja Jürgen Schmidhuberin vuonna 1997 esittämä long short-term memory (LSTM) on tähän yksi mahdollinen ratkaisu (Hochreiter & Schmidhuber 1997).

LSTM:n käyttämät muistisolut, jotka kulkevat sen yksiköiden läpi, oppivat tunnistamaan tärkeät tapahtumat syöttäen niitä uudelleen soluun, kunnes tapahtuma ei ole enää relevantti ja se voidaan unohtaa (Gers ym. 1999). Prosessi tapahtuu yksikössä porttien avulla, joista tieto kulkee. Portit säätelevät solun tilaa, joka määrää edellisten tapahtumien vaikutuksen ulostuloon. Toisin sanoen solu oppii muistamaan tulevaisuuden kannalta hyödylliset tapahtumat eikä jokaista yksityiskohtaa (Olah 2015; van Houdt ym. 2020, 3–5). Ihmisen aivojen muistamisen toimintaperiaate on samankaltainen (Géron 2022), ja LSTM onkin kyennyt ihmismäisiin tehtäviin (Greff ym. 2016; van Houdt ym. 2020, 7–19). LSTM:n tarkempi toimintaperiaate selitetään luvussa 4.

Opinnäytetyön rajaaminen kosteaan silmänpohjan ikärappeumaan mahdollistaa potilaan tilan kehittymisen havaitsemisen. Jokaisella hoito- ja seurantakäynneillä tallennetaan potilaan tiedot, kuten silmän mittaukset ja mahdollisen lääkepidoksen antamisen. Näistä voidaan havaita potilaan tilan kehittyminen parempaan tai huonompaan suuntaan edelliseen käyntikertaan nähden. Tämä antaa mahdollisuuden potilaan tilan ennustamiseen seuraavalle käyntikerralle koneoppimismenetelmällä, kun otetaan huomioon potilaan perustiedot, riskitekijät, hoitotoimenpiteet ja tutkimusmittaukset käyntikerroilla sekä niiden väliset aikajaksot. Tautiin on esimerkiksi tehty kuvien perusteella ennustamistutkimusta koneoppimismenetelmällä (Bhuiyan ym. 2020).

Opinnäytetyön tarkoitus oli kokeilla, onko LSTM-keinoverkon hyödyntäminen mahdollista synteettiseen kostean silmänpohjan ikärappeumapotilasdataan. Synteettinen eli keinotekoinen aineisto on luotu aidon aineiston pohjalta. Työssä aluksi avataan sairautta yleisellä tasolla käymällä läpi olennaisimmat asiat koneoppimisen näkökulmasta ja sen jälkeen perehdytetään neuroverkkoihin. LSTM esitellään luvussa 4, ja luvussa 5 esitetään LSTM-mallin kokeilun tuloksia, ja lopuksi tehdään työn yhteenveto sekä jatkotutkimusehdotukset.

2 Silmänpohjan ikärappeuma

Silmänpohjan tai verkkokalvon ikärappeuma (AMD, age-related macular degeneration) eli makuladegeneraatio on ikääntyneiden yleisin tarkan näkemisen alueelle kohdistuva silmätauti heikentäen näköä vakavasti (Komulainen ym. 2016). Tautia on kahta muotoa, hitaasti etenevä kuiva ja nopeasti etenevä nesteinen tai kostea silmänpohjan ikärappeuma (wAMD, wet AMD). wAMD:n ilmiöön ei tiedetä varsinaista syytä. Taudissa silmänpohjan suonikalvoston verisuonet kasvavat hallitsemattomasti tihkuen nestettä ympäristöön, aiheuttaen turvotusta ja verkkokalvon eri kerrosten irtoamista toisistaan. (Seppänen 2021.)

Vaikka molemmissa tautimuodoissa riskitekijät ja oireet ovat samanlaisia, kuivaan rappeumaan ei kuitenkaan ole parantavaa hoitokeinoa vielä. Tähän voidaan vaikuttaa ainoastaan elämäntapamuutoksilla, kuten ruokavalioon ja riskitekijöiden poistamiseen tai vähentämiseen. Potilaan tilaa seurataan 1–2 vuoden välein. (Kaarniranta 2018, 192–93.)

2.1 Oireet

Sairauteen kuuluvat näkökykyyn liittyvät oireet, joita ovat lähi- ja kaukonäön häiriöt, värinäön ja tarkan näön heikkeneminen sekä viivojen vääristyminen ja tumman varjostuman havaitseminen keskeisessä näkökentässä (Kaarniranta 2018, 192). Nesteisessä muodossa oireet ilmenevät päivien tai viikkojen aikana, kun taas kuivassa oireet etenevät vuosissa (Komulainen ym. 2016; Kivinen ym. 2017, 641–642). Kostean silmänpohjan ikärappeuman syynä on silmänpohjan suonikalvoston vuotavien uudisverisuonten muodostuminen verkkokalvolle aiheuttaen veren ja kudoksen nesteen vuotoa tarkan näön alueelle, josta tulee taudin nimitys (Kaarniranta & Kivinen 2021).

2.2 Tärkeimmät riskitekijät

Suurin riskitekijä on ikääntyminen, kuten taudin nimikin viittaa ikääntymisen aiheuttamaan rappeutumiseen. Ikääntymisen myötä verkkokalvon rakenteelliset ja toiminnalliset muutokset altistavat taudille. Mitä korkeampi ikä sitä korkeampi riski on sairastua tautiin. (Heesterbeek ym. 2020, 147.)

Tietyt perityt geenit ja niiden variantit lisäävät riskiä sairastua tautiin (Genet 2016). Perimän osuus taudin ilmaantuvuudesta on noin 65 %. Se on toiseksi merkittävin riskitekijä ikääntymisen lisäksi, jota ei voida ehkäistä. Esimerkiksi kaikkien kolmen geenin, tekijä H:n, LOC387715:n ja C3:n, kantajalla on 18-kertainen riski sairastaa tauti verrattuna ei-kantajaan (Seitsonen ym. 2009, 2361–2362).

Tupakointi lisää sairastumisen riskiä 2–4-kertaiseksi kumpaan tahansa silmänpohjan ikärappeumaan. Vaikka tupakointiin voidaan vaikuttaa sen lopettamisella, on lopettaneilla silti kohtuullinen riski sairastua tautiin. Lopettaminen joka tapauksessa vähentää riskiä ja 20 vuoden tupakoimattomuuden jälkeen riski laskee ei-tupakoitsijoiden tasolle (Heesterbeek ym. 2020, 148.)

Sydän- ja verisuonitaudit lasketaan riskitekijöihin, joita ovat mm. kohonnut verenpaine, valtimonkovettumatauti ja suurentuneet kolesteroliarvot. Lihavuus lisää näiden riskiä (Mustajoki 2021). Näihin tavallisesti liittyvät vähäinen liikkuminen ja huonot ruokatottumukset. (Kaarniranta 2018, 192; Seppänen 2021.) Tärkeimpien riskitekijöiden taustalla voivat vaikuttaa myös muut riskitekijät, joita ei tässä opinnäytetyössä ole otettu huomioon. Näistä on kattavammin kirjoitettu kirjallisuuskatsauksessa. (Heesterbeek ym. 2020.)

2.3 Tutkiminen

Potilasta tutkitaan mittaamalla näöntarkkuutta biomikroskooppitutkimuksella, silmänpohjan kuvaamisella. Silmänpohjan kuvantamisen apuna voidaan käyttää fluoreseiiniväriainetutkimusta (FAG), jolla saadaan poikkeavat

verisuonilöydökset selville. Väriaine paljastaa verkkokalvon alla kasvaneet uudissuonet ja verkkokalvon turvotusalueet. Tarvittaessa käytetään täydentävää indosyaniinivihreäväriainetutkimusta (ICG). (Kivinen ym. 2018, 1239.)

Verkkokalvon valokerroskuvaamista (OCT) käytetään yleisimmin verkkokalvosairauksissa, jolla saadaan verkkokalvon poikkileikkauskuva ja kolmiulotteisuus selville. Kuvista voidaan selvittää verkkokalvon pienetkin paksuusmuutokset, jotka vaikuttavat hoidon aloittamiseen. (Kivinen ym. 2018, 1239–1241.)

2.4 Hoito ja seuranta

Laser- eli valoaktivaatiohoitoa (PDT) käytetään vain yksittäisissä tapauksissa, koska sen tehokkuus riippuu tautimuodosta. Tutkielmassa tämä on jätetty huomiotta, koska eniten käytetty ja vallitseva hoitomuoto on lasiaisinjektiot. (Seppänen 2021.)

Lasiaisinjektiolla, jossa verisuonikasvutekijöitä (VEGF) annostellaan silmän lasiaiseen, pyritään estämään uudissuonimuodostuksia ja hidastetaan taudin etenemistä sekä näön heikkenemistä (Kaarniranta & Kinnunen 2014, 2266–2268). Hoidon aloittaminen tarpeeksi ajoissa parantaa taudin ennustetta (Seppänen 2021). Käytettäviä VEGF:iä ovat aflibersepti, bevasitsumabi ja ranibitsumabi (Kuoppala & Härkönen 2015, 20–27).

VEGF-hoitoprotokollia on kolme. Suunnitellusti säännöllisin väliajoin annosteltava (fixed), jossa ensimmäiset kolme lääkepistoa annetaan 4–6 viikon hoitovälein, ja seurantatutkimukset suoritetaan pistoskertojen välissä. Afliberseptin kohdalla 4–6 viikon jälkeen lääkepistojen väli on 8 viikkoa. (Komulainen ym. 2016.)

Tarvittaessa annosteltava (pro re nata, PRN) annetaan afliberseptia 4–8 viikon ja muilla lääkkeillä 4–6 viikon välein hoitovasteen ollessa riittämätön. Seurantatutkimukset tehdään viimeisen suunnitellusti annetun pistoksen

jälkeen, ja hoitovasteen oltaessa riittävä ei anneta pistosta ja seurantatutkimus tehdään 4–6 viikon päästä. Tähän voidaan suunnitella kolmen pistoksen sarja, jolloin seurantatutkimus tehdään tämän lopussa. (Komulainen ym. 2016.)

Treat and extend (TAE) on tilanteen mukaan vähitellen pidentyvin väliajoin annosteltava hoitomuoto. Ennen riittävää hoitovasteen saavuttamista ovat hoitovälit samat kuin PRN:ssa. Seuranta tehdään aina ennen suunniteltua pistosta. Kun riittävä hoitovaste on saavutettu, seurantavälit harvennetaan korkeintaan 2 viikkoa kerrallaan ja korkeintaan 12 viikon väleihin. Havaitessa puutteellinen hoitovaste palataan edelliseen hoitoväliin. (Komulainen ym. 2016.)

3 Neuroverkot

Neuroverkot (NN, neural network) on yksi koneoppimisen (ML, machine learning) menetelmä ja koneoppiminen on tekoälyn (AI, artificial intelligence) alaluokka. Ihmisen älykkyys perustuu kokemuksen myötä saavutettavaan asiantuntemukseen eli toisin sanoen koneoppiminen on kuin ihmisen kokemus tekoälylle, jotta se voisi tehdä päätöksiä ja ratkaista ongelmia. (Shalev-Shwartz & Ben-David 2014, 19–22.) Teoreettisesti tekoälyllä tarkoitetaan konetta, jolla on älykkyys tehdä ihmismäisiä tehtäviä (IBM Cloud Education 2020b).

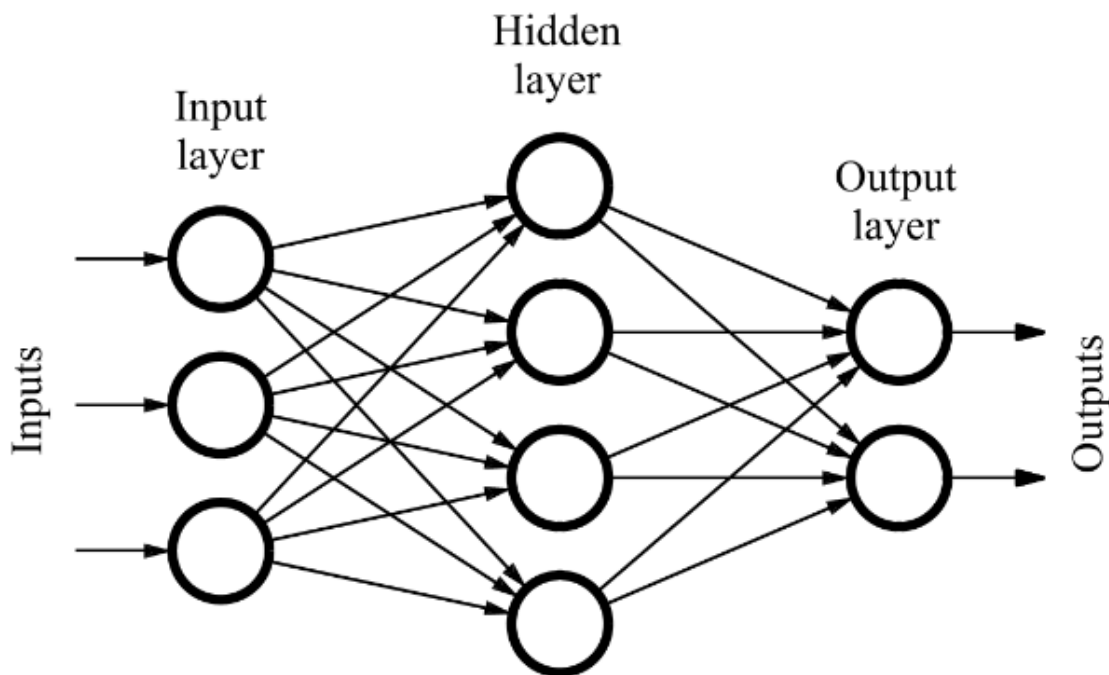
Koneoppiminen jaetaan pääasiassa valvottuun (supervised learning), valvomattomaan (unsupervised learning) ja vahvistusoppimiseen (reinforcement learning) (IBM Cloud Education 2020d). Tässä tutkielmassa olennaisin on ensimmäinen, jossa on käytetty luokiteltua (labeled) dataa. Kyseisessä tapauksessa kone tietää syötteen (x , features) ja ulostulon (y , target) muuttujat. Koneoppimismallin valvoja säätää mallia, kunnes haluttu ulostulo on saavutettu eli tällä pyritään yleistämään funktionaalisia x :n ja y :n välisiä toimintamalleja. (Shalev-Shwartz & Ben-David 2014, 22–23.)

Valvomattomassa koneoppimisessä ulostulo on tuntematon ja malli oppii itseksensä tuntemattomia rakenteita, malleja ja ryhmiä syötetyllä luokittelemattomalla (unlabeled) datalla etsimällä sen yhtäläisyyksiä. Vahvistusoppimisessä oppiva agentti on vuorovaikutuksessa ympäristönsä kanssa, josta se saa positiivista tai negatiivista palautetta riippuen, onko se onnistunut tai epäonnistunut halutussa suoritteessa. Agentin tavoitteena on maksimoida palkkio. Hyvä esimerkki tästä on mallin oppiminen pelien pelaamiseen. (Mohri ym. 2018, 379–380; Sutton & Barto 2020, 1–13.)

Neuroverkon tai keinoneuroverkon (ANN, artificial neural network) inspiraation lähteenä on ihmisen aivojen neuroverkot (Mcculloch & Pitts 1943). Keinoneuroverkoissa on useita neuroneita, kuten ihmistenkin neuroverkoissa, jotka ovat yksittäisiä laskentayksiköitä. Ne ovat kytkettyinä toisiinsa mahdollistaen tiedonkulun neuronien välillä. Tämä mahdollistaa

monimutkaistenkin laskutoimitusten suorittamisen. (Shalev-Shwartz & Ben-David 2014, 268–269; Géron 2022.)

Keinoneuroverkkoja on monenlaisella arkkitehtuurilla. LSTM:n kannalta tärkeimmät ovat myötäkytketty neuroverkko (FNN, feedforward neural network) (Kuvio 1), sekä takaisinkytketty neuroverkko (RNN, recurrent neural network).

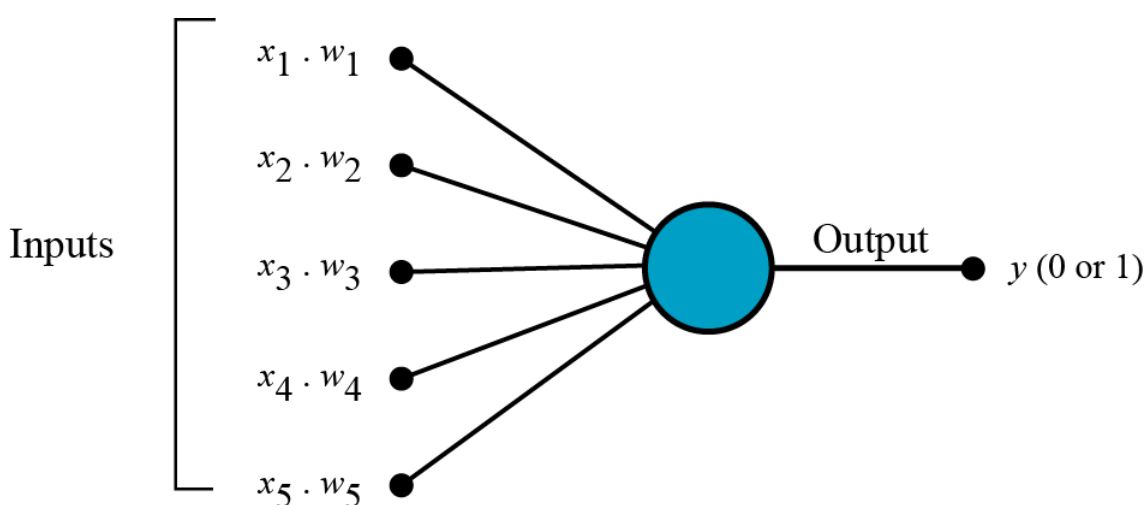


Kuvio 1. Myötäkytketty neuroverkko (Davim & Quiza 2011)

3.1 Neuroni

Neuroni tai solmu (neuron, node) on käytännössä pieni laskentasuoritin, jonka tehtävänä on säädellä tiedonkulkua aktivoitumalla ja linkittymällä muihin neuroneihin, kuten aivojenkin neuronit (Alpaydin 2016, 85–91). ANN-arkkitehtuureissa neuronit ovat järjestetty kerrokseen L (layer). Kuvio 1 mukaisesti tulo- ja ulostulon välissä olevia kerroksia kutsutaan piilokerrokseksi (hidden layer), joita voi olla mielivaltaisen määrä. Nämä auttavat konetta ymmärtämään $x:n$ ja $y:n$ välisen sidonnaisuuden. Jos arkkitehtuurissa on useampi piilokerros tulo- ja ulostulokerroksen välissä käytetään nimitystä syväoppiminen (DL, deep learning), mikä viittaa neuroverkon kerroksien syvyyteen. (Heaton 2015, 8.)

Neuroni tai perseptroni on alun perin Frank Rosenblattin (1958) kehittämä ensimmäinen ja yksinkertainen neuroverkko. Tämä vastaanottaa linkkien avulla edellisen kerroksen neuroneilta tulevia syötteitä Kuvio 2 mukaisesti x_1 , x_2 , x_3 , x_4 ja x_5 , joilla on omat painokertoimensa w (weight). Syötteitä voi olla mielivaltaisen määrä. Laskemalla nämä yhteen painokertoimillaan saadaan painotettu summa eli lineaarinen regressio, joka mittaa syötteen tärkeyden, mikä taas määrää neuronin aktivoitumisen (activation) ja sen tason. Siten neuroni saa ulostulonsa, y :n, aktivaatiofunktioistaan eli toisin sanoen kyseessä on epälineaarinen eli logistinen regressio. (Nielsen 2015; Géron 2022.)

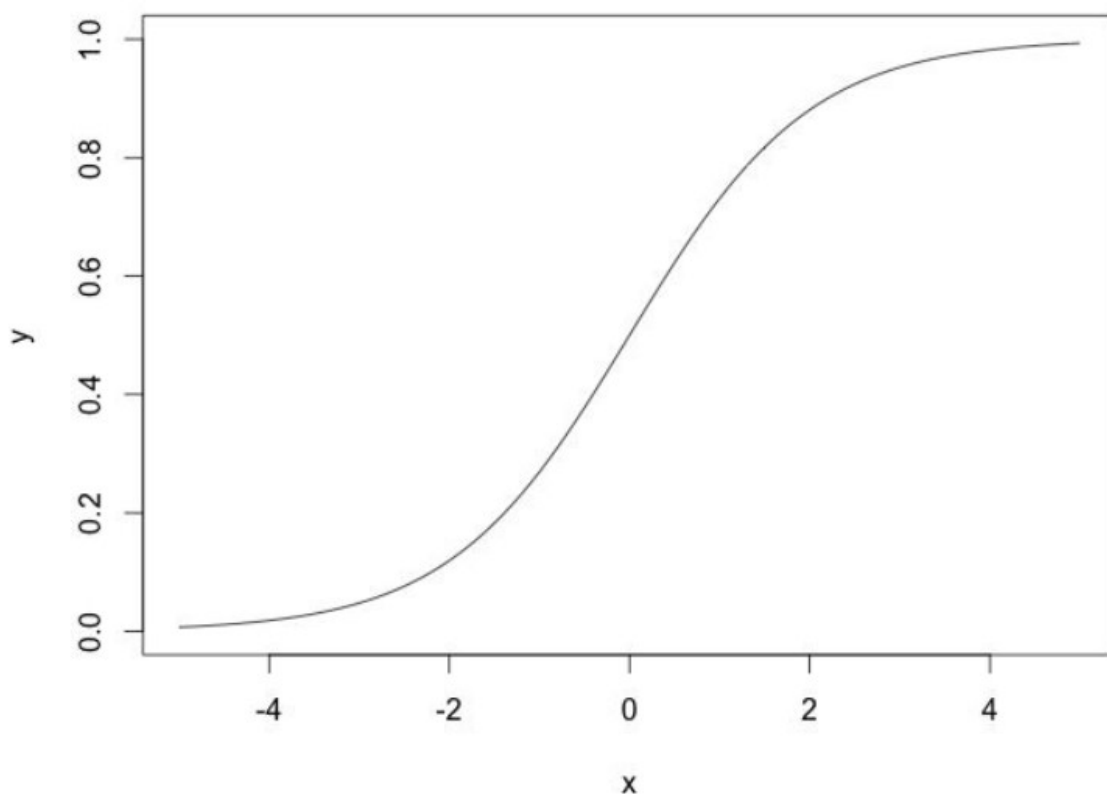


Kuvio 2. Perseptroni viidellä sisääntulolla ja yhdellä ulostulolla (Sharma 2017)

Neuronin aktivaatiofunktioita on olemassa useampia. Tässä opinnäytetyössä keskitytään vakioon LSTM:ään, joka käyttää vakiona sigmoidi- ja hyperbolista tangenttiaktivaatiofunktioita.

Sigmoidifunktio

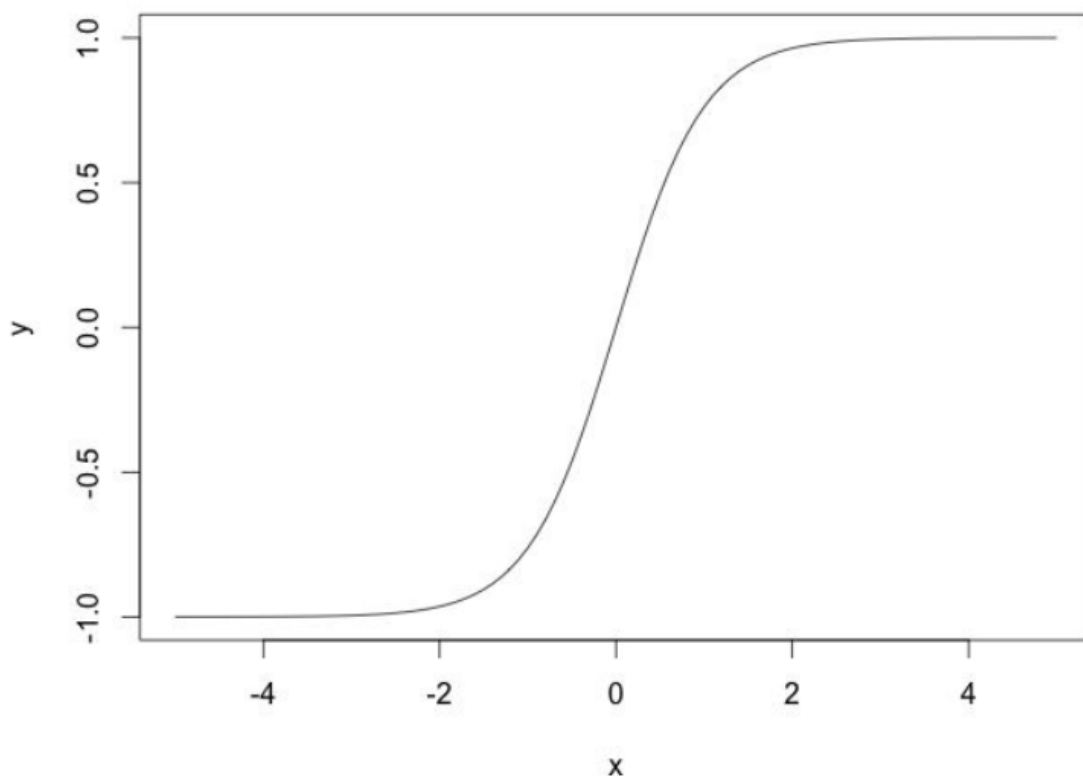
Sigmoidifunktio on yleisin aktivaatiofunktio, joka pakottaa neuronin ulostulon positiiviseksi numeroksi. Kuten Kuvio 3 nähdään, sen ulostulo y on aina nollan ja yhden välillä riippumatta siitä, kuinka suuri sisääntulo x on.



Kuvio 3. Sigmoidi aktivaatiofunktio (Heaton 2015, 14)

Hyperbolinen tangenttifunktio

Kuvio 4 voidaan havaita hyperbolisella tangenttifunktiolla olevan samantyyppinen kuvaaja kuin sigmoidifunktiolla. Kuitenkin tämän funktion ulostulo y on aina välillä -1 ja 1 . Vahvasti positiivinen sisääntulo x merkitään positiiviseksi (1) ja vahvasti negatiivinen sisääntulo x merkitään negatiiviseksi ulostuloksi (-1).



Kuvio 4. Hyperbolinen tangenti aktivaatiofunktio (Heaton 2015, 15)

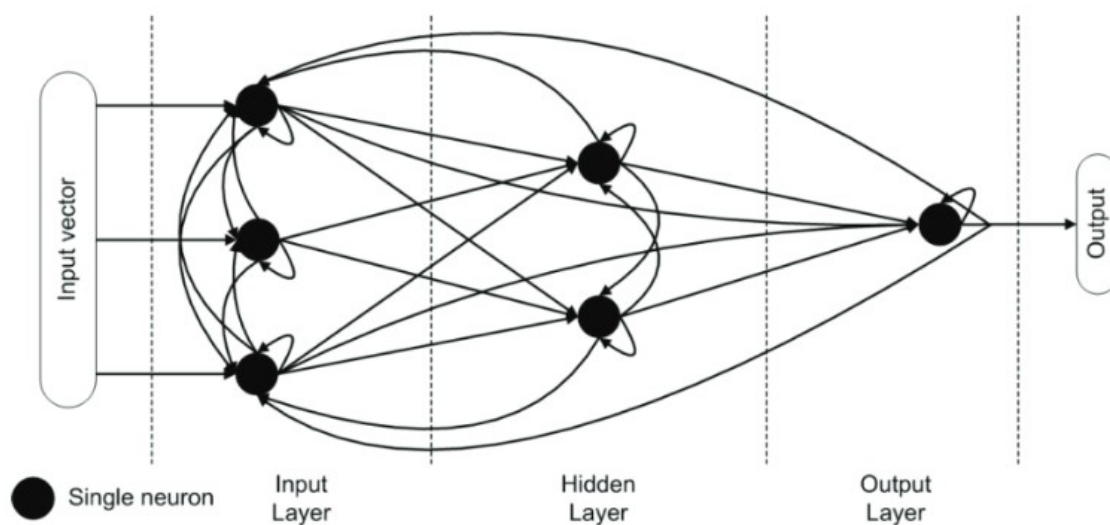
3.2 Myötäkytketty neuroverkko

Kuvio 1 oleva neuroverkko on myötäkytketty neuroverkko (FNN, feedforward neural network), jossa tieto kulkee sisääntulosta eteenpäin piilokerrosten läpi ulostulokerrokseen. Viimeisen kerroksen funktiona on normalisoiva aktivaatiofunktio, esimerkiksi logistinen sigmoidi- (Kuvio 3) tai hyperbolinen tangenttifunktio (Kuvio 4), joka tuottaa ulostulokerroksen neuronien ulostulon. (Goodfellow ym. 2016, 191–192).

3.3 Takaisinkytketty neuroverkko

Takaisinkytketyssä neuroverkossa (RNN, recurrent neural network) on vähintään yksi syklinen polku, joka tarkoittaa kerroksen ulostulon kytkemistä takaisin saman tai edellisen kerroksen neuronin sisääntuloon (Kuvio 5). Tämä

luo verkolle sisäisen muistitilan, joka vaikuttaa tulevan oppimiseen (Krenker ym. 2011, 8).



Kuvio 5. Täysin takaisinkytketty neuroverkko (Krenker ym. 2011)

Sisään tuleva jaksoittainen tieto, esimerkiksi virke on kytkeytynyt edellisiin tai tuleviin sanoihin. Neuroverkko analysoi virkkeen sana kerrallaan, ja jotta se ymmärtäisi asiayhteyden, pitää sen syöttää edellinen sana takaisin verkkoon ymmärtääkseen seuraavan sanan. Ajan hetket pätevät myös, kun ihminen lukiessaan muistaa, mitä edellisessä lauseessa kirjoitettiin. (Goodfellow ym. 2016, 367–368.) Esimerkkinä mainittakoon Google Kääntäjän neurojärjestelmä (Wu ym. 2016).

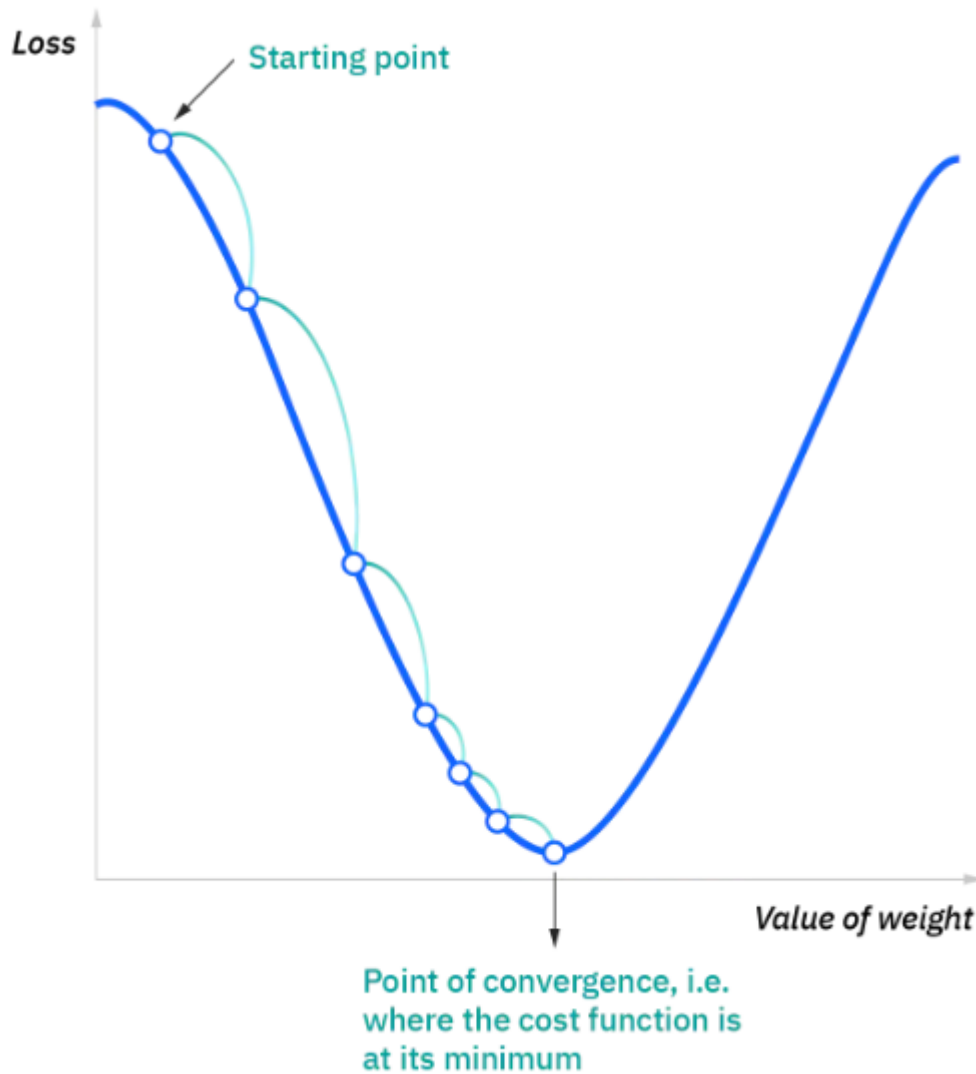
Mainittakoon, että näiden lisäksi on olemassa takaisinkytkettyjen neuroverkkojen erikoistapauksia kuten molemmat suunnat huomioon ottava kaksisuuntainen eli bidirektionaalinen neuroverkko (BRNN, bidirectional recurrent neural network) (Schuster & Paliwal 1997) sekä kuvien ja kuvioiden tunnistamiseen ja konenäköratkaisuihin käytetty konvoluutioinen neuroverkko (CNN, convolutional neural network) (Das ym. 2019). Neuroverkkoja voidaan myös yhdistää (Nguyen ym. 2022).

3.4 Katoavan gradientin ongelma

Neuroverkon oppiessa, tappiofunktio (loss function) laskee tappioksi (loss) ulostulon eron haluttuun ulostuloon, josta saadaan mallin tarkkuus (accuracy) esitettyä prosentteina. Tappion minimoimisella pyritään saamaan mallista mahdollisimman optimaalinen. (Amari 2021, 13–14.)

Neuroverkkojen käyttämät painokertoimet ovat opittavia parametreja, jotka auttavat neuroverkkoa yhdistämään aiemmin opitun tiedon nykyiseen tehtävään. Painot säätelevät neuronien välisten yhteyksien vahvuudet, joten neuroverkon oppiessa se päivittää painot jokaisella iteraatiolla niin, että se luo neuronille parhaan ulostuloarvon, joka tuottaa paremman mallin tarkkuuden. Tämä mahdollistaa edellisen kerroksen painojen muutoksien riippuvuuden seuraavan kerroksen painoihin. Toisin sanoen se yrittää oppia $x:n$ ja $y:n$ välisen suhteen iteroimalla. (Nielson 2022.)

Yleisesti neuronien painojen päivittämiseen käytetään gradientteja, jotka saadaan funktion osittaisderivoinnilla. Ne arvioivat useiden muuttujien funktion pienimmän ulostulon ottaen huomioon sen suurimman muutosnopeuden ja muutoksen suunnan. (Khan Academy; Adams & Essex 2018, 724.) Gradientteja pitää laskea alas (gradient descent) niin monta kertaa (epoch), kunnes tappiofunktion ulostuloksi saadaan mahdollisimman alhainen arvo (IBM Cloud Education 2020c; Géron 2022). Toisin sanoen se yrittää luoda ja päivittää funktion sellaiseksi, joka luo $x:n$ ja $y:n$ välisen suhteen mahdollisimman tarkasti oikein. Kuvio 6 avulla voidaan helpommin kuvitella pallon eli funktion vierivän alas tason syvimpään kohtaan lyhintä reittiä. Tämän laskeminen takaisinkytketyissä neuroverkoissa tapahtuu yleensä vastavirta-algoritmileillä (backpropagation) (Bengio ym. 1993, 1183–1184; Géron 2022).



Kuvio 6. Gradienttilasku (Amari 2021)

Vastavirta-algoritmi laskee tappiofunktion gradientteja aloittaen aikasarjan viimeisestä kerroksesta ja edeten kohti ensimmäistä neuronikerrosta päivittäen neuronien painoja eli aikaisempaa aika-askelta (time step). Koska ensimmäisten kerrosten gradientit lähestyvät nollaa ja painot ovat usein neuronin aktivaatiofunktiolla rajoitettu välille 0–1, useammalla aika-askeleella ja pidemmällä aikasarjoilla ensimmäisten kerrosten painot eivät päivity. Tämä estää takaisinkytketyn neuroverkon oppimisen. (Bengio ym. 1994; Hochreiter ym. 2003.) Tämä ilmiö on nimeltään katoavan gradientin ongelma (Hochreiter 1998). Esimerkiksi LSTM on erikoistunut pitkien aikavälien riippuvuuksiin.

4 LSTM-neuroverkko

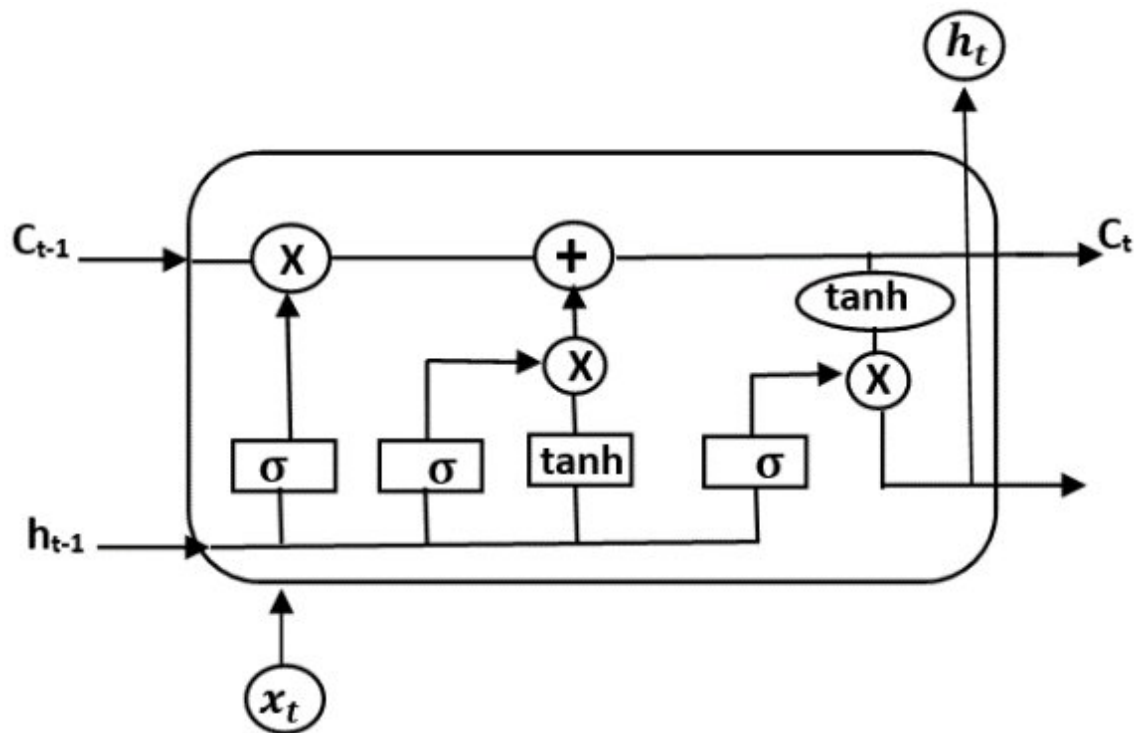
LSTM-neuroverkko kehitettiin ratkaisemaan katoavan gradientin ongelman (Hochreiter & Schmidhuber 1997). Se on takaisinkytketty neuroverkko, mutta takaisinkytkentöjen lisäksi se käyttää tavallisten neuronien sijaan muistisoluja, jotka käytännössä ovat itse takaisinkytkettyjä alineuroverkkoja. Muistisolu säätelee informaation kulkemista solun porteilla (gate) ja solun tilalla (cell state), joka mahdollistaa pitkien aikavälien riippuvuuksien huomioon ottamiseen, mikä tekee LSTM:n erikoiseksi. (Ravichandiran 2018, 149–150; van Houdt ym. 2020, 3–4.) Näin LSTM voi oppia itse kokemuksensa perusteella tärkeitä tapahtumia, vaikka niiden välillä olisi pitkiä tuntemattoman suuria aikaviiveitä (Krenker 2011, 10–11).

4.1 LSTM-muistisolu

Muistisolun (Kuvio 7) sisääntuloina toimivat edellisen aika-askeleen solun tila (C_{t-1}) ja ulostulon piilotila (h_{t-1} , hidden state), sekä nykyisen aika-askeleen sisääntulo (x_t) (Ravichandiran 2018, 150). Piilotilalla tarkoitetaan esimerkiksi lauseen seuraavan ajanhetken ($t + 1$) kontekstuaalisen sanan ennustamista aikaisempien ajanhetkien sanojen sisääntulojen eli muistettavien asioiden perusteella (IBM Cloud Education 2020a). Sanoja ei yleensä tulosteta ulos yksitellen jokaisella aika-askeleella, vaan tarkoitus on nähdä koko ennustettu ulostulo kokonaisuutena lauseena tai lauseina (y). Tässä opinnäytetyössä käytetään ulostulon merkinä piilotilaa h ulostulon y :n sijaan, sillä molempia on käytetty ristiin viitatuissa teoksissa (Olah 2015; Ravichandiran 2018; van Houdt ym. 2020). Tämän lisäksi van Houdt:n (2020) teoksessa h on merkitty ulostuloportin aktivaatiofunktioiksi.

Solussa h_{t-1} ja x_t kulkevat kolmen portin ja niiden aktivaatiofunktioiden, sigmoidin (σ) ja hyperbolisen tangentin (\tanh) (ks. luku 3.1) läpi, jotka määräävät porttien ulostulot. Nämä taas vaikuttavat nykyisen aika-askeleen solun tilaan (C_t) ja ulostuloon (h_t). Kaikki nuolet esittävät vektoreita,

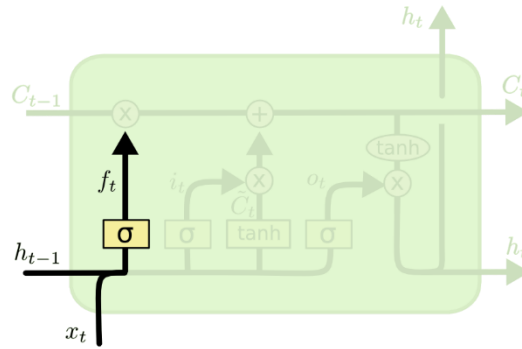
mahdollistaen solun tilan vektorin alkioiden unohtamisen, päivittämisen tai säilyttämisen. (Ravichandiran 2018, 150; van Houdt ym. 2020, 4.) Jokainen Kuvio 7. LSTM-muistisolun rakenne (Cai ym. 2019) Kuvio 7 aktivaatiofunktioilla varustettu suorakulmainen laatikko kuvastaa oppinutta neuroverkon kerrosta ja ympyrän muotoiset alkioita operaattoreita (Olah 2015).



Kuvio 7. LSTM-muistisolun rakenne (Cai ym. 2019)

Unohtusportti

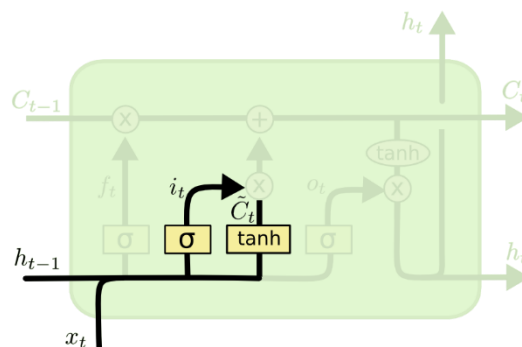
Ensimmäinen portti on unohtamiseen tarkoitettu portti (forget gate) (Kuvio 8). Sen tehtävä on määrätä mitä tietoa poistetaan viime aika-askleen solun tilasta, ettei turha informaatio vaikuttaisi nykyiseen tilaan (Ravichandiran 2018, 150). Tämän portin aktivaatioon, joka on aina sigmoidifunktio, vaikuttavat nykyisen sisääntulon ja viime aika-askleen ulostulon yhdistelmä (Greff ym. 2016, 2; van Houdt ym. 2020, 4–5).



Kuvio 8. Unohdusportti (Olah 2015)

Sisääntulo- ja päivitysportti

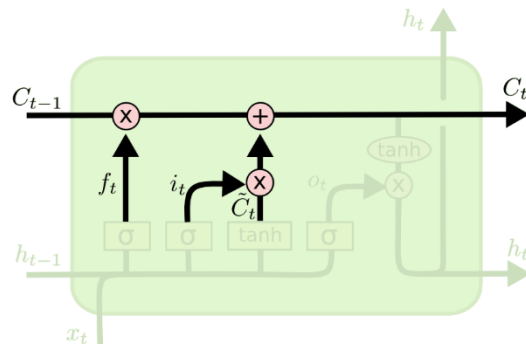
Kuvio 9 voidaan nähdä, että solun tilaa päivitetään samoilla sisääntuloilla kuin edellisellä portilla. Aina sigmoidifunktiolla varustetun sisääntuloportin (input gate) päättäessä päivitettävät arvot (i_t), ja tavallisesti hyperbolisella tangenttifunktiolla varustettu päivitysportti (update gate) luo vektorin uusilla kandidaattiarvoilla eli uuden nykyisen tilan ehdokkaan (\tilde{C}_t). Seuraavaksi nämä kaksi yhdistetään kertomalla alkioittain. (Ravichandiran 2018, 151; van Houdt ym. 2020, 4.)



Kuvio 9 Sisääntulo- ja päivittämisportti (Olah 2015)

Viime aika-askeleen solun päivitys tapahtuu kaksivaiheisesti, ensiksi kertomalla se alkioittain unohdusportin tuotoksella (f_t), ja sen jälkeen alkioittain lisäämällä alkioittain kerrotun sisääntulo- ja päivitysportin tulon (Kuvio 10). Lopputuloksena

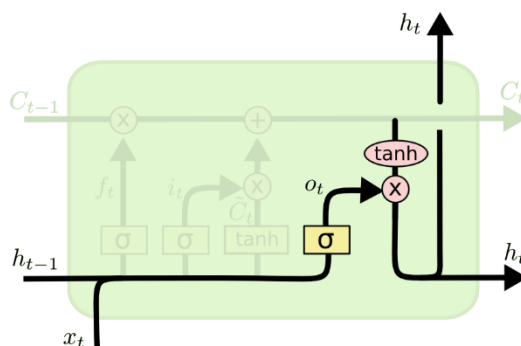
saadaan nykyisen ajanhetken solun tila seuraavalle aika-askelelle.
(Ravichandiran 2018, 151; van Houdt ym. 2020, 4–5.)



Kuvio 10 Solun tilan päivittäminen (Olah 2015)

Ulostuloportti

Viimeiseen porttiin tulevat samat sisääntulot niin kuin edellisissäkin kulkien sigmoidin kautta, päättäen mitkä alkioit otetaan solun tilasta ulostuloon (o_t) (Kuvio 11). Näiden lisäksi syötetään nykyisen solun tila alkioittain kerrottuna hyperbolisella tangentilla mukaisesti, pakottaen niiden arvot olemaan $-1:n$ ja $1:n$ välillä. Lopuksi nämä kerrotaan yhteen alkioittain nykyisen aika-askleen ulostuloksi, joka myös siirtyy nykyisen solun tilan kanssa seuraavan aika-askleen muistisolulle. Toisin sanoen ulostulon perusteena on suodatettu nykyisen ajan hetken solun tila, jossa päätettiin, mitkä relevantit muistettavat osat otetaan ulostuloon. Solun tila ulostulon kanssa yhdessä liikkeessa muistisolusta seuraavaan aika-askleen muistioluun mahdollistaa neuroverkkoa muistamaan kauan aikaa sitten opitut tärkeät asiat.
(Ravichandiran 2018, 151; van Houdt ym. 2020, 5.)



Kuvio 11 Ulostuloportti (Olah 2015)

4.2 LSTM:n variaatiot

Aivan ensimmäinen versio LSTM:stä oli ilman unohdusporttia. Se lisättiin siihen melko nopeasti solun tilan nollaamiseen estääkseen solun tilan lineaarista kasvua monella aika-askeleella rikkoen neuroverkon (Gers ym. 1999). Heti tämän jälkeen lisättiin kurkistusaukot (peephole), joilla syötetään myös solun tila jokaiseen porttiin ennen solun tilan muokkaamista (Gers & Schmidhuber 2000). Tämän on kuitenkin havaittu lisäävän laskennallista raskautta eikä sen tarkkuus eronnut merkittävästi alkuperäisen LSTM:n tarkkuuteen (Breuel 2015).

LSTM-muistisolusta on myös kehitetty yksinkertaisempi kaksiporttinen, päivittämis- ja resetoitiportilla varustettu versio GRU (gated recurrent unit), käyttäessään edellisen aika-askeleen aktivaatiota solun tilan sijaan (Cho ym. 2014). Chung ym. (2014) on havainnut GRU:n suorituskyvyn olevan verrattavissa LSTM:n suorituskykyyn ainakin puheen ja monisinfonisen musiikin signaalien mallintamiseen. Muuten LSTM:n variaatioilla ei ole havaittu olevan parempaa suoriutumista verrattuna alkuperäiseen LSTM:ään (Greff ym. 2016).

Ajan myötä LSTM:stä on kehitetty erilaisia variaatioita, jotka ovat menestyneet erilaisissa käyttötarkoituksissa. Tässä opinnäytetyössä kuitenkin keskitytään unohdusportilla varustettuun alkuperäiseen LSTM:ään (Vanilla LSTM). Lisäksi on huomioitava, että eri suhteellisen tuoreissakin artikkeleissa Vanilla LSTM:n muistisolulla voidaan tarkoittaa kurkistusaukoilla (van Houdt ym. 2020, 4) tai ilman (Nigri ym. 2019, 6).

5 Ennustuskokeilu ja tulokset

Luvussa käydään läpi valvotun koneoppimisen ja LSTM:n käytännön osuutta ohjelmointitasolla niin, että lukija ymmärtää sisällön aiempien teorialukujen avulla. Tämä luku sisältää, mitä tarvittavia työvälineitä ja kirjastoja tarvitaan valvottuun koneoppimisprosessiin, Vanilla LSTM:n ohjelmointiin ja tuloksien esittämiseen graafisesti.

5.1 Ohjelmointityövälineet ja -kirjastot

Käytännön osuuden resurssien säästämiseksi Googlen Colaboratory tai Colab osoittautui käteväksi vapaaksi alustaksi, jonka pohjana on Jupyter-muistikirjapalvelu. Tämä mahdollistaa käytön ilman erillisiä asennuksia, koska koneoppimiseen tarvittavat kirjastot ja työvälineet ovat valmiina käytettäviksi. Ohjelmointikielenä käytetään Python 3:a. Colabiin kirjaudutaan Google-tilillä selaimella, jolloin palvelinta voi käyttää etänä. Jupyter-muistikirjat tallentuvat pilvipalveluun Google Driveen, jota käyttää myös esimerkiksi Googlen Gmail-sähköposti. (Google Colab.)

Ensimmäinen ja tärkein vaihe valvotussa koneoppimismenetelmässä on raakadatan esikäsittely. Tällä tarkoitetaan datan muokkaamista niin, että koneoppimismallin olisi helpompi käsitellä dataa tarkkuuden parantamiseksi. Valvojan luettavuuden vuoksi datan sisällön muokkaaminen järkevämpään muotoon edesauttaa virhetilanteiden käsittelyä ja myöhempää tuloksien analyttistä tulkintaa. Valvotussa koneoppimisessä raakadata yleensä jaetaan noin 70–80 % opetusdataan ja 20–30 % validointia varten testidataan, joissa molemmissa ovat x ja y -setit (ks. luku 3). Tässä työssä käytettiin suhdetta 67:33, joka saatiin sklearn:in eli Scikitin kirjastosta (Pedregosa ym. 2011) saatavalla olevalla `train_test_split`-työvälineellä (Kuvio 12). Tähän jakosuhteeseen päädyttiin siksi, että aineisto oli suuri ja tällä pyrittiin välttämään koneoppimismallin opitun tiedon liiallista yleistämistä validointidataan.

Kuvio 12 on kaikki kirjastot, joita tässä tutkimuksessa käytettiin. Pandas- ja Numpy-kirjastoja käytetään esimerkiksi raakadatan lukemiseen, analysointiin ja muokkaamiseen (Mckinney 2010; Harris ym. 2020). TensorFlow (Abadi ym. 2015) on avoimen lähdekoodin alusta, joka on tarkoitettu koneoppimiseen ja sen rajapintana toimii Keras (Chollet ym. 2015), joka sisältää syväoppimiseen tarkoitettun neuroverkkokirjaston. Matplotlibia ja mlxtendia käytettiin visuaaliseen esittämiseen ja analysointiin (Hunter 2007; Raschka 2018). Scikit-Learnin, Kerasin ja TensorFlow'n kirjastoista ja niiden käytöstä Python-ohjelmointikielellä löytyy teoksesta (Géron 2022).

Kuvio 12

```
[1] from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np
from numpy import array
import tensorflow as tf
from tensorflow import keras
from keras.models import Model, Sequential
from keras.layers import LSTM, Dense
import matplotlib.pyplot as plt
from mlxtend.plotting import plot_confusion_matrix
```

Kuvio 12 Käytetyt kirjastot ja työvälineet

5.2 Vanilla LSTM-mallin rakentaminen

Kuvio 13 on esitettyä yksinkertainen LSTM neuroverkkoarkkitehtuuri Python-ohjelmoinnilla rakennettuna. Siinä on sisääntulokerros parametreina 24 yksikköä ja vastaanottaen opetusdatan ulottuvuuden muodon. Piilokerros 8 yksiköllä ja ulostulokerros (ks. luku 3). Dense tarkoittaa neuroverkkokerrosta, joka on kytketty jokaiseen edellisen kerroksen yksikköön. Shape kuvastaa opetusdatan ulottuvuuden elementtien määrää. Esimerkiksi kaksiulotteisessa datassa ovat elementteinä rivit ja sarakkeet. Neuroverkon parametreja ja kerroksien määrää voidaan muuttaa mielivaltaisesti. Niitä kuitenkin pyritään

muokkaamaan niin, että mallista tulisi mahdollisimman tarkka riippuen vastaanotettavasta datasta.

```
[52] model = Sequential() # neuroverkkomallin alustaminen
      model.add(LSTM(24, input_shape=(X_raw_train.shape[1], X_raw_train.shape[2]))) # sisääntulokerros (aika-askeleet, ominaisuudet)
      model.add(Dense(8)) # piilokerros
      model.add(Dense(y_raw_train.shape[1])) # ulostulokerros
      model.compile(optimizer='adam', loss='mse', metrics=['accuracy']) # mallin konfigurointi
```

Kuvio 13 Vanilla LSTM

5.2.1 Käytetty tappiofunktio

Tappiofunktioita on useita eri datasisältöjä varten (ks. luku 3.4). Tässä työssä tappiofunktiona käytettiin keskimääräistä neliövirhettä (MSE, mean squared error) (Sammut & Webb 2010). Funktion palauttama arvo eli keskimääräinen neliöero lasketaan kunkin ulostuloneuronin arvioitujen ja todellisen arvon välillä. Tämä näyttää kuinka väärässä neuroverkko on ennustettaessa haluttua ulostuloa.

5.2.2 Käytetty optimoija

Optimointialgoritmin tai -funktion tehtävänä on muokata neuroverkon painoja ja oppimisnopeutta vähentäen mallin tappiota ja kasvattaen tarkkuutta (Gupta 2021). Optimointialgoritmina käytettiin Adamia (Adaptive Moment Estimation), koska se laskee ja päivittää yksilölliset adaptiiviset oppimisnopeudet jokaiselle verkon parametreille eli painoille vain ensimmäisen ja toisen gradientin hetken arvoista vähällä muistin tarpeella (Kingma & Ba 2014, 1; Géron 2022) Adamia suositellaankin käyttämään vakiona optimoijana sen tehokkuutensa takia useammassa koneoppimisongelmassa (Kingma & Ba 2014). Gradienttilaskussa voidaan tämän kuvitella olevan mäkeä laskeutuva painava pallo kitkalla (Ruder 2016).

5.3 Vanilla LSTM-mallin oppiminen

Kuvio 14 voidaan nähdä mallin oppimisen prosentuaalinen tappion ja tarkkuuden etenemisen opetusdatalla, mallin nähdessään sen 10 kertaa (ks. luku 3.4).

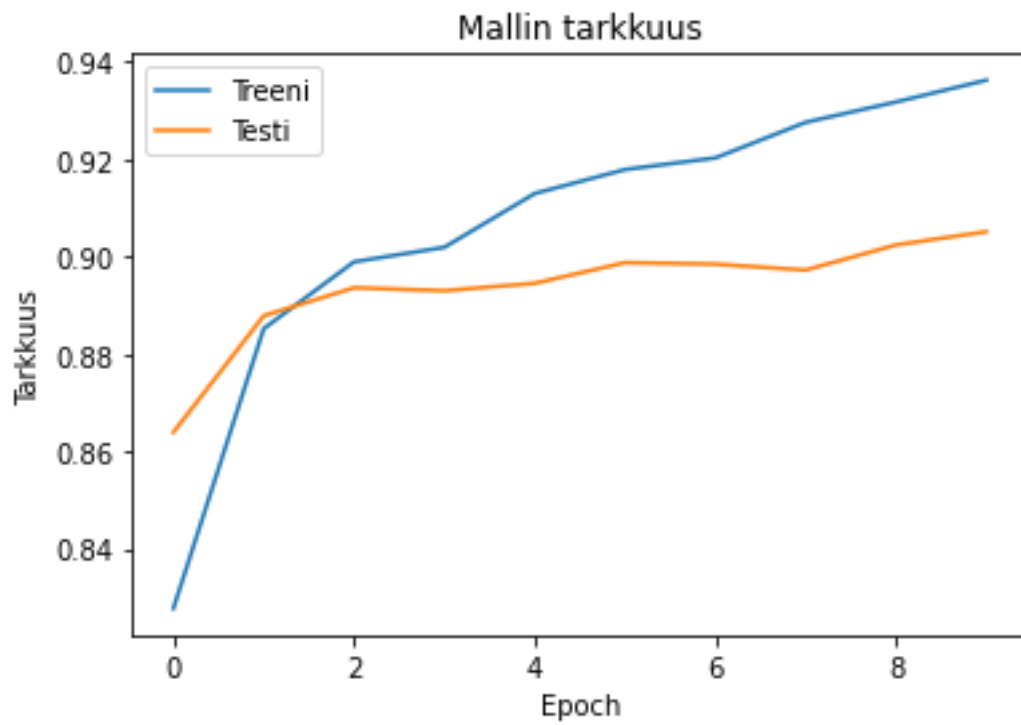
Mallille vielä näkemätön validointidata on myös mukana vertailussa, jotta nähdään mallin todellinen tappio ja tarkkuus (val_loss, val_accuracy) eli toisin sanoen mallin sopivuus uuteen dataan. Mallin mittarit tallennetaan muuttujaan history, jotta mallin graafien luominen olisi mahdollista.

```
[261] history = model.fit(X_raw_train, y_raw_train, epochs=10, validation_data=(X_raw_test, y_raw_test) )

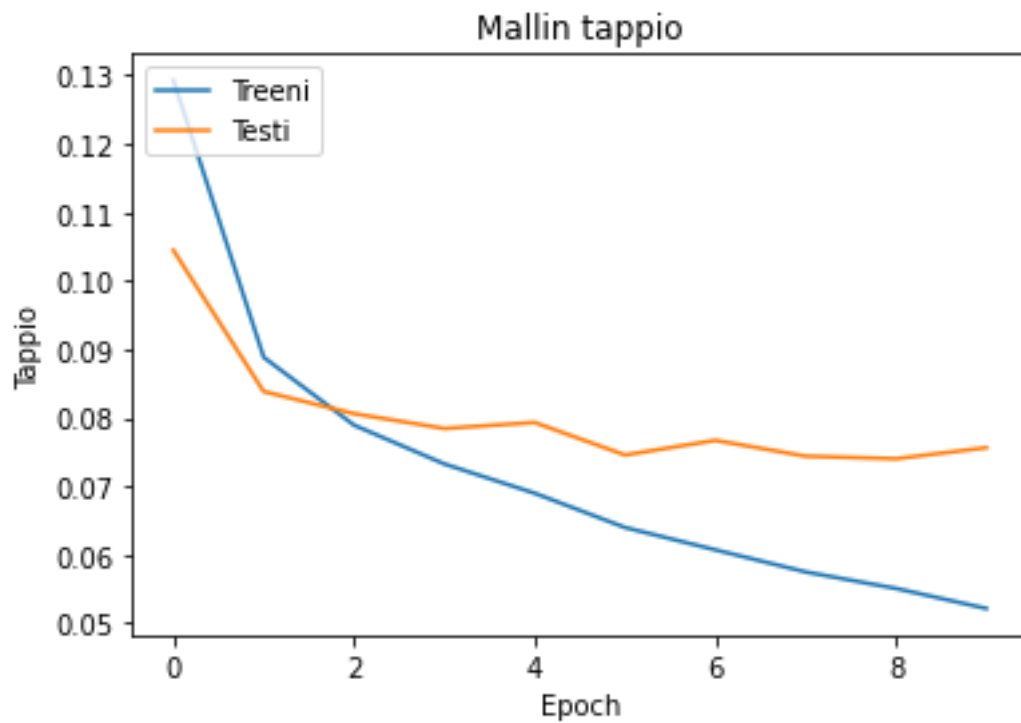
Epoch 1/10
210/210 [=====] - 5s 8ms/step - loss: 0.1293 - accuracy: 0.8278 - val_loss: 0.1045 - val_accuracy: 0.8639
Epoch 2/10
210/210 [=====] - 1s 5ms/step - loss: 0.0888 - accuracy: 0.8852 - val_loss: 0.0838 - val_accuracy: 0.8879
Epoch 3/10
210/210 [=====] - 1s 6ms/step - loss: 0.0789 - accuracy: 0.8990 - val_loss: 0.0806 - val_accuracy: 0.8936
Epoch 4/10
210/210 [=====] - 1s 5ms/step - loss: 0.0733 - accuracy: 0.9019 - val_loss: 0.0784 - val_accuracy: 0.8930
Epoch 5/10
210/210 [=====] - 1s 5ms/step - loss: 0.0690 - accuracy: 0.9130 - val_loss: 0.0793 - val_accuracy: 0.8945
Epoch 6/10
210/210 [=====] - 1s 5ms/step - loss: 0.0640 - accuracy: 0.9179 - val_loss: 0.0746 - val_accuracy: 0.8988
Epoch 7/10
210/210 [=====] - 1s 6ms/step - loss: 0.0607 - accuracy: 0.9203 - val_loss: 0.0767 - val_accuracy: 0.8985
Epoch 8/10
210/210 [=====] - 1s 5ms/step - loss: 0.0575 - accuracy: 0.9276 - val_loss: 0.0744 - val_accuracy: 0.8973
Epoch 9/10
210/210 [=====] - 1s 6ms/step - loss: 0.0551 - accuracy: 0.9318 - val_loss: 0.0740 - val_accuracy: 0.9024
Epoch 10/10
210/210 [=====] - 1s 5ms/step - loss: 0.0522 - accuracy: 0.9363 - val_loss: 0.0756 - val_accuracy: 0.9052
```

Kuvio 14 Vanilla LSTM-mallin oppiminen

Kuvio 15 voidaan nähdä, että mallin nähdessä saman opetusdatan useamman kerran tieto yleistyy ja tarkkuus kasvaa. Yleistämäänsä tietoaan se käyttää validointidataan, mikä ilmaistaan testitarkkuutena. Vastaavanlaisesti toimii mallin tappion esitys, jolloin tappio laskee jokaisella näkemiskerralla kohti nollaa (Kuvio 16). Kuvio 16 voidaan myös havaita, ettei malli ylisovitu (overfitting) eikä alisovitu (underfitting) (Belkin ym. 2019, 2–3; IBM Cloud Education 2021). Validointidatalla tarkkuus sekä tappio saattavat vaihdella enemmän, koska se ei ole nähnyt validointidataa aiemmin.



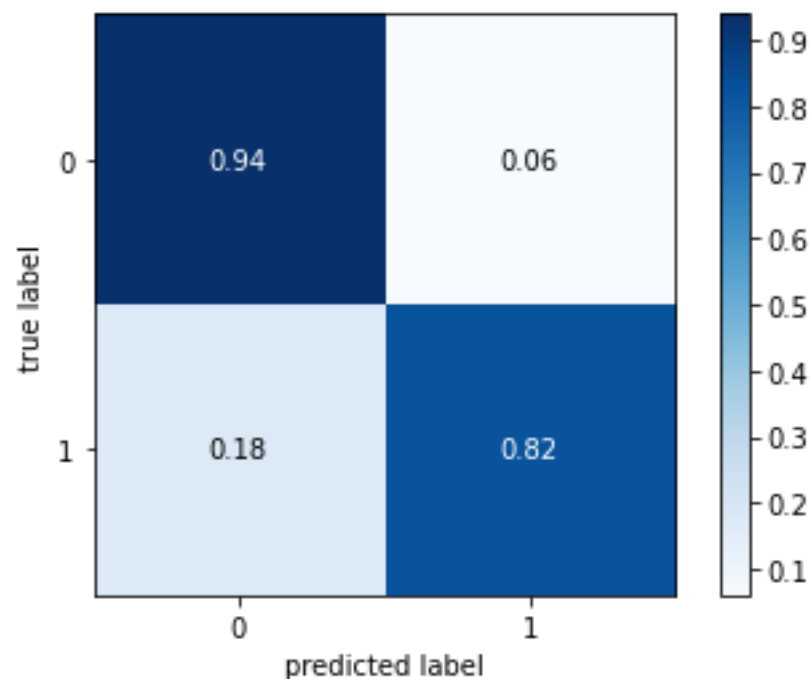
Kuvio 15 Vanilla LSTM-mallin tarkkuus



Kuvio 16 Vanilla LSTM-mallin tappio

5.4 Vanilla LSTM-mallin tulokset

Koneoppimismallin ennustamisprosessi tapahtuu validointidatan (x) pohjalta. Mallin ennustamistarkkuutta mitataan hämmennys- tai virhematriisilla (confusion matrix). Tämä perustuu mallin luokittelutarkkuuteen eli siihen, kuinka monta A-luokkaa malli luokitteli B-luokaksi kaikilla luokkien A:n ja B:n pareilla (Géron 2022). Tässä tapauksessa verrataan ennustettuja y:n arvoja tai luokkia (predicted label) todellisiin y:n arvoihin tai luokkiin (true label) (Kuvio 17). Tässä tutkielmassa potilaan tilaa ennustettaessa ilmaistaan, onko potilaan tila jatkossa huono (luokka 0) eli tarvitaan hoidon tehostamista tai hyvä (luokka 1) eli ei vaadi lisätoimia. Voidaan havaita, että ennustettuja huonotilaisia malli ennusti 94-prosenttisesti ja parempitilaisia 82-prosenttisesti oikein.



Kuvio 17 Vanilla LSTM-mallin confusion matrix prosentteina

6 Lopuksi

Opinnäytetyön tavoitteena oli kokeilla long short-term memory-koneoppimismenetelmän soveltuvuutta isoon synteettiseen kostean silmänpohjan ikärappeumapotilasdataan. LSTM on erikoistunut pitkiin aikaväleihin, joita voi olla potilasdatassa useita. Synteettisellä datalla pyrittiin ennustamaan potilaiden tilan muuttumista parempaan tai huonompaan suuntaan seuraavaan käyntikertaan asti. Tämän perusteella voidaan tehdä päätös, voiko LSTM-menetelmää soveltaa alkuperäiseen potilasdataan.

Työssä käytiin teoriaa läpi tärkeimmistä asioista silmänpohjan nesteisestä ikärappeumasta ja sen yleisyydestä. Tämän lisäksi otettiin huomioon sairauden etenemiseen vaikuttavat merkittävimmät seikat. Nämä tiedot potilasdatassa luovat pohjan koneoppimisen mahdollistamiseen. Koneoppimista ja neuroverkkoja selvitettiin vähintään siitä näkökulmasta, että koneoppiminen, neuroverkkojen toimintaperiaate sekä niiden peruskäsitteet olisivat ymmärrettäviä. LSTM-menetelmän toimintaperiaate läpi yksityiskohtaisesti, jota käytettiin synteettiseen potilasdataan kokeillakseen sen soveltuvuutta yleisesti. Käytännön osuudessa käytettiin Python-ohjelmointia synteettisen datan esikäsittelyyn LSTM-neuroverkolle sopivaksi, LSTM-neuroverkon rakentamiseen ja tuloksien graafiseen esittämiseen.

Tulokseksi saatiin koneoppimismenetelmän kannalta merkittävä ennustustarkkuus mallille. Esimerkiksi Banerjee ym. (2020) saivat konvoluutio-LSTM:llä kostean AMD:n erittymisriskin ennustamisessa kuvadatan perusteella samankaltaisia tarkkuuksia. Havaitut samankaltaiset ennustetarkkuudet koskevat lähiajan eli kolmen kuukauden ennustetarkkuuksia, sillä kauemmaksi ennustettaessa mallin tarkkuus laskee, mutta käyntikertojen lisäämisen myötä se nousi.

Saatujen tuloksien perusteella voitaisiin kokeilla LSTM-mallia suoraan alkuperäiseen dataan, koska synteettinen data oli kelvollinen koneoppimiseen. Tämä voisi helpottaa lääkärin työtä, kun ennustetaan potilaan tila seuraavaan käynti- tai hoitokertaan. Tällöin lääkäri osaa varautua seuraavaan hoitokäyntiin.

ja tarvittaessa muuttaa hoitoprotokollaa. Jos tässä työssä käytetty malli ei suoriudu halutulla tavalla, on neuroverkkoarkkitehtuurin muokkaus suhteellisen helppoa ja nopeaa, ja se tapahtuu mallin hyperparametrien ja neuroverkkokerroksien muokkaamisella (Géron 2022).

Opinnäytetyö oli hyvin rajoittunut sairauden ja koneoppimismenetelmän kannalta. Sairauden kohdalla otettiin huomioon vain merkittävimmät riskitekijät. Koneoppimismenetelmän kannalta käytettiin vain yksinkertaista LSTM:ää. Samaa dataa voisi kokeilla LSTM-variaatioilla ja mahdollisesti ottaa sairauden kaikki riskitekijät huomioon.

Koska LSTM:ää on käytetty päätöksentekovälineenä, jatkokehittämisen kohteena olisi Vanilla LSTM:n ja mahdollisesti sen variaatioiden hyödyntäminen lääkkeen valitsemiseen potilaalle ottaen huomioon, mitä lääkettä on annettu aiemmin ja mihin suuntaan potilaan tila on kehittynyt (Sipilä ym. 2018b). Tähän voitaisiin ottaa mukaan lääkeaine brolusitsumabi, joka on saanut myyntiluvan Euroopassa vuonna 2020 (Härkönen & Kiviniemi 2020, 5). Tämän lisäksi aiempien tutkimusten tueksi (Rein ym. 2009; Wong ym. 2014) voitaisiin LSTM:n aikasarjojen hyödynnettävyyden avulla ennustaa lääkkeiden myyntimäärien ja niiden kustannusten kehitykset esimerkiksi vuosiin 2040–2050 mennessä. (Sipilä ym. 2018a; Härkönen & Kiviniemi 2020, 12–13.) Kolmas jatkokehitys edellä mainittujen pohjalta voitaisiin ennustaa eri lääkevaihtoehtojen kustannustehokkuutta (Rafferty ym. 2007).

Lähteet

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. Viitattu 26.4.2022.

<https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45166.pdf>.

Adams, R.A. & Essex, C. 2018. Calculus: A Complete Course. 9th ed. Toim. O'Donnell, C., Rancourt, L. & Morrison-Reed, C. Don Mills, Ontario, Canada: Pearson Canada Inc. 1168 s. ISBN 978-0-13-415436-7.

Alpaydin, E. 2016 MACHINE LEARNING - The New AI. Cambridge, London, England: The MIT Press. 224 s. (The MIT Press Essential Knowledge Series). ISBN 978-0262529518.

Amari, A. 2021. A Beginner's Guide to Deep Learning. *IBM Think 2021*. IBM Corporation 2021. S. 1–16 Viitattu 20.4.2022.

<https://community.ibm.com/HigherLogic/System/DownloadDocumentFile.ashx?DocumentFileKey=2f18fc13-d575-9e6a-7a1d-dc611d0cb676&forceDialog=0>

Baji, T. 2018. Evolution of the GPU Device widely used in AI and Massive Parallel Processing. IEEE 2nd Electron Devices Technology and Manufacturing Conference (EDTM) 2018. Kobe, Japan. 13-16.3.2018. IEEE. S. 7–9. Viitattu 11.4.2022. DOI: 10.1109/EDTM.2018.8421507.

Banerjee, I., de Sisternes, L., Hallak, J. A., Leng, T., Osborne, A., Rosenfeld, P. J., Gregori, G., Durbin, M. & Rubin, D. 2020. Prediction of age-related macular degeneration disease using a sequential deep learning approach on longitudinal SD-OCT imaging biomarkers, *Scientific Reports*. Vol. 10, 1, 1–16. Viitattu 27.4.2022. DOI: 10.1038/s41598-020-72359-y.

Belkin, M., Hsu, D., Ma, S. & Mandal, S. 2019. Reconciling modern machine learning practice and the bias-variance trade-off. Ohio, Columbus, USA: The Ohio State University; New York, USA: Columbia University. Viitattu 27.4.2022. DOI:10.48550/arXiv.1812.11118.

Bengio, Y., Frasconi, P. & Simard, P. 1993. The Problem of Learning Long-Term Dependencies in Recurrent Networks. IEEE International Conference on Neural Networks. San Francisco, CA, USA. 28.3-1.4.1993. 1183–1188. Viitattu 12.4.2022. DOI:10.1109/ICNN.1993.298725.

https://www.researchgate.net/publication/224663746_Problem_of_learning_long-term_dependencies_in_recurrent_networks.

Bengio, Y., Simard, P. and Frasconi, P. 1994. Learning Long-Term Dependencies with Gradient Descent is Difficult. IEEE Transactions on Neural Networks. Vol. 5, No 2, 157–166. Viitattu 12.4.2022. DOI:10.1109/72.279181. https://www.researchgate.net/publication/5583935_Learning_long-term_dependencies_with_gradient_descent_is_difficult.

Bhuiyan, A., Wong, T.Y., Ting, D.S.W., Govindaiah, A., Souied, E.H. & Smith, R.T. 2020. Artificial Intelligence to Stratify Severity of Age-Related Macular Degeneration (AMD) and Predict Risk of Progression to Late AMD. Translational Vision Science & Technology. Vol. 9, No 2, Article 25, 1–25. Viitattu 7.2.2022. DOI:10.1167/TVST.9.2.25.

Breuel, T.M. 2015. Benchmarking of LSTM Networks. Viitattu 21.4.2022. DOI:10.48550/arxiv.1508.02774.

BrightFocus Foundation 2021. Age-Related Macular Degeneration: Facts & Figures. Macular Degeneration Research. Viitattu 3.4.2022. <https://www.brightfocus.org/macular/article/age-related-macular-facts-figures>.

Cai, R., Li, S., Tian, J. & Ren, L. 2019. Short-term Load Forecasting Based on Electricity Price in LSTM in Power Grid. IOP Conference Series: Materials Science and Engineering. Vol. 569, No 4. Viitattu 21.4.2022. DOI:10.1088/1757-899X/569/4/042046.

Cho, K. *et al.* 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. Proceedings of SSST 2014 - 8th Workshop on Syntax, Semantics and Structure in Statistical Translation. S. 103–111. Viitattu 21.4.2022. DOI:10.48550/arxiv.1409.1259.

Chollet, F. & ym. 2015. Keras. Viitattu 4.26.2022. <https://github.com/keras-team/keras>.

Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. NIPS 2014 Deep Learning and Representation Learning Workshop. Viitattu 21.4.2022. DOI:10.48550/arxiv.1412.3555.

Das, A., Rad, P., Choo, K.R., Nouhi, B., Lish, J. & Martel, J. 2019. Distributed machine learning cloud teleophthalmology IoT for predicting AMD disease progression. Future Generation Computer Systems. Vol. 93, 486–498. Viitattu 11.4.2022. DOI:10.1016/J.FUTURE.2018.10.050.

Davim, J.P. & Quiza, R. 2011. Computational Methods and Optimization. Teoksessa: J.P. Davim (toim.) Machining of Hard Materials. Berlin, Germany: Springer-Verlag, 177–208. Viitattu 7.4.2022. DOI:10.1007/978-1-84996-450-0. https://www.researchgate.net/publication/234055177_Computational_Methods_and_Optimization.

Fritsche, L.G., Igl, W., Cooke Bailey, J.N., Grassmann, F., Sengupta, S., Bragg-Gresham, J.L., Burdon, K.P., Hebring, S.J., Wen, C., Gorski, M., Kim, I.K., Cho, D., Zack, D., Souied, E., Scholl, H.P.N., Bala, E., Lee, K.E., Hunter, D.J., Sardell, R.J., Mitchell, P., Merriam, J.E., Cipriani, V., Hoffman, J.D., Schick, T., Lechanteur, Y.T.E., Guymer, R.H., Johnson, M.P., Jiang, Y., Stanton, C.M., Buitendijk, G.H.S., Zhan, X., Kwong, A.M., Boleda, A., Brooks, M., Gieser, L., Ratnapriya, R., Branham, K.E., Foerster, J.R., Heckenlively, J.R., Othman, M.I., Vote, B.J., Liang, H.H., Souzeau, E., McAllister, I.L., Isaacs, T., Hall, J., Lake, S., Mackey, D.A., Constable, I.J., Craig, J.E, Kitchner, T.E., Yang, Z., Su, Z., Luo, H., Chen, D., Ouyang, H., Flagg, K., Lin, D., Mao, G., Ferreyra, H., Stark, K., von Strachwitz, C.,N, Wolf, A., Brandl, C., Rudolph, G., Olden, M., Morrison,

M.A., Morgan, D.J., Schu, M., Ahn, J., Silvestri, G., Tsironi, E.E., Hyung Park, K., Farrer, L.A., Orlin, A., Brucker, A., Li, M., Curcio, C., Mohand-Saïd, S., Sahel, J-A., Audo, I., Benchaboune, M., Cree, A.J., Rennie, A., Goverdhan, S.V., Grunin, M., Hagbi-Levi, S., Campochiaro, P., Katsanis, N., Holz, F.G., Blond, F., Blanché, H., Deleuze, J-F., Igo, R.P. Jr., Truitt, B., Peachey, N.S., Meuer, S.M., Myers, C.E., Moore, E.L., Klein, R., Hauser, M.A., Postel, E.A., Courtenay, M.D., Schwartz, S.G., Kovach, J.L., Scott, W.K., Liew, G., Tfan, A.G., Gopinath, B., Merriam, J.C., Smith, R.T., Khan, J.C., Shahid, H., Moore, A.T., McGrath, J.A., Laux, R., Brantley, M.A. Jr., Agarwal, A., Ersoy, L., Caramoy, A., Langmann, T., Saksens, N.T.M., de Jong, E.K., Hoyng, C.B., Cain, M.S., Richardson, A.J., Martin, T.M., Blangero, J., Weeks, D.E., Dhillon, B., van Duijn, C.M., Doheny, K.F., Romm, J., Klaver, C.C.W., Hayward, C., Gorin, M.B., Klein, M.L., Baird, P.N., den Hollander, A.I., Fauser, S., Yates, J.R.W., Allikmets, R., Wang, J.J., Schaumberg, D.A., Klein, B.E.K., Hagstrom, S.A., Chowers, I., Lotery, A.J., Léveillard, T., Zhang, K., Brilliant, M.H., Hewitt, A.W., Swaroop, A., Chew, E.Y., Pericak-Vance, M.A., DeAngelis, M., Stambolian, D., Haines, J.L., Iyengar, S.K., Weber, B.H.F., Abecasis, G.R. & Heid, I.M. 2016. A large genome-wide association study of age-related macular degeneration highlights contributions of rare and common variants. *Nature Genetics*. Vol. 48, No 2, 134–143. Viitattu 8.2.2022. DOI:10.1038/ng.3448. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4745342/>.

Géron, A. 2022. *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow, Concepts, Tools and Techniques to Build Intelligent Systems*. 3rd ed. Toim. R. Novack, M. Cronin, & B. Kelly. Sebastopol, California: O'Reilly Media, Inc. Viitattu 10.5.2022. <https://www.oreilly.com/library/view/hands-on-machine-learning/9781098125967/>.

Gers, F.A. & Schmidhuber, J. 2000. Recurrent nets that time and count. *Teoksessa: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*. Como, Italy. 27-27.7.2000. IEEE. Vol. 3, S. 189–194. ISBN:0-7695-0619-4. Viitattu 21.4.2022.

DOI:10.1109/IJCNN.2000.861302.

https://www.researchgate.net/publication/3857862_Recurrent_nets_that_time_and_count.

Gers, F.A., Schmidhuber, J. & Cummins, F. 1999. Learning to forget: Continual prediction with LSTM. *Neural Computation*. Vol. 12, No 10, 2451–2471. MIT Press Journals. Viitattu 21.4.2022. DOI:10.1162/089976600300015015.

https://www.researchgate.net/publication/12292425_Learning_to_Forget_Continual_Prediction_with_LSTM.

Goodfellow, I., Bengio, Y. & Courville, A. 2016. *Deep Learning*. MIT Press. 781 s. Viitattu 11.4.2022. <https://www.deeplearningbook.org/>.

Google Colab. n. d. Viitattu 26.4.2022.

<https://research.google.com/colaboratory/faq.html>.

Greff, K. Srivastava, R.K., Koutnik, J., Steunebrink, B.R., & Schmidhuber, J., 2016. LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*. IEEE. Vol. 28, No. 10, 2222–2232. Viitattu 21.4.2022. DOI:10.1109/TNNLS.2016.2582924.

Gupta, A. 2021. *A Comprehensive Guide on Deep Learning Optimizers*. Data Science Blogathon. Analytics Vidhya. Viitattu 27.4.2022.

<https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/>.

Härkönen, U. & Kiviniemi, V. 2020. *Brolusitsumabi*. EUnetHTA-raportin yhteenveto ja kustannusten arviointi. Lääkealan turvallisuus- ja kehittämiskeskus Fimea. 18 s. Viitattu 2.4.2022.

https://www.fimea.fi/documents/160140/1454401/Brolusitsumabi_yhteenveto.pdf.

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Rio, J.F., Wiebe, M., Peterson, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke,

C. & Oliphant, T.E. 2020. Array programming with NumPy. *Nature* 2020. Vol. 585, No. 7825, 357–362. Viitattu 26.4.2022. DOI:10.1038/s41586-020-2649-2.

Heaton, J. (toim.). 2015. Artificial intelligence for humans, Volume 3: Deep Learning and Neural Networks. St. Louis, Missouri, USA: Heaton Research, Inc. 374 s. ISBN:978-1505714340. Viitattu 6.4.2022.

<https://www.heatonresearch.com/book/aifh-vol3-deep-neural.html>.

Heesterbeek, T.J., Lor Es-Motta, L., Hoyng, C.B., Lechanteur, Y.T.E. & Den Hollander, A.I. 2020. Risk factors for progression of age-related macular degeneration. *Ophthalmic Physiol Opt.* John Wiley & Sons, Inc. Vol, 40, No. 2, 140–170. Viitattu 7.2.2022. DOI:10.1111/opo.12675.

Hochreiter, S. 1998. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems.* Vol. 6, No. 2, 107–116. Viitattu 7.4.2022. DOI:10.1142/S0218488598000094.

Hochreiter, S., Bengio, Y., Frasconi, P. & Schmidhuber, J. 2003. Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies.

Teoksessa: S.C. Kremer & J.F. Kolen (toim.). *A Field Guide to Dynamical Recurrent Neural Networks.* Wiley-IEEE press. S. 237–243. Viitattu 12.4.2022.

<https://www.researchgate.net/publication/2839938>.

Hochreiter, S & Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation.* Vol. 9, No. 8, 1735–1780. Viitattu 6.4.2022.

DOI:10.1162/neco.1997.9.8.1735.

https://www.researchgate.net/publication/13853244_Long_Short-term_Memory.

Van Houdt, G., Mosquera, C. & Nápoles, G. 2020. A review on the long short-term memory model. *Artificial Intelligence Review.* Vol. 53, No. 1, 5929–5955.

Viitattu 13.4.2022. DOI:10.1007/S10462-020-09838-1.

https://www.researchgate.net/publication/340493274_A_Review_on_the_Long_Short-Term_Memory_Model.

Hunter, J.D. 2007. Matplotlib: A 2D graphics environment. *Computing in Science and Engineering*. IEEE. Vol. 9, No. 3, 90–95. Viitattu 26.4.2022. DOI:10.1109/MCSE.2007.55.

IBM Cloud Education. 2020a. What are Recurrent Neural Networks? IBM Cloud Learn Hub. Viitattu 22.4.2022. <https://www.ibm.com/cloud/learn/recurrent-neural-networks>.

IBM Cloud Education. 2020b. What is Artificial Intelligence (AI)? IBM Cloud Learn Hub. Viitattu 5.4.2022. <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>.

IBM Cloud Education. 2020c. What is Gradient Descent? IBM Cloud Learn Hub. Viitattu 20.4.2022. <https://www.ibm.com/cloud/learn/gradient-descent>.

IBM Cloud Education. 2020d. What is Machine Learning? IBM Cloud Learn Hub. Viitattu 5.4.2022. <https://www.ibm.com/cloud/learn/machine-learning>.

IBM Cloud Education. 2021. What is Overfitting? IBM Cloud Learn Hub. Viitattu 27.4.2022. <https://www.ibm.com/cloud/learn/overfitting>.

Jørstad, Ø.K., Steffensen, L.A., Eriksen, K., Bragadóttir, R. & Moe, M.C. 2020. Thirteen years of intravitreal anti-vascular endothelial growth factor therapy: the promises and burdens of a paradigm shift told from the perspective of the largest retina service in Norway. *Acta Ophthalmologica*. John Wiley & Sons, Inc. Vol. 98, No. 8, 774–779. Viitattu 3.4.2022. DOI:10.1111/AOS.14177.

Kaarniranta, K. 2018. Silmänpohjan ikärappeumat. Teoksessa: Uusitalo, H. (toim.). *Silmätautien käsikirja*. 1. painos. Helsinki: Kustannus Oy Duodecim, 192–194. ISBN:978-951-656-422-0.

Kaarniranta, K. & Kinnunen, K. 2014. Silmänpohjan kostea ikärappeuma pitää hoitaa ajoissa. *Suomen Lääkärilehti*. Vol. 37, No. 69, 2265–2268a. Viitattu 1.4.2022. <https://docplayer.fi/7601362-Silmanpohjan-kostea-ikarappeuma-pitaa-hoittaa-ajoissa.html>.

Khan Academy. n. d. Gradient descent. Khan Academy. Viitattu 21.4.2022.
<https://www.khanacademy.org/math/multivariable-calculus/applications-of-multivariable-derivatives/optimizing-multivariable-functions/a/what-is-gradient-descent>.

Kingma, D.P. & Ba, J.L. 2014. Adam: A Method for Stochastic Optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. San Diego, California, USA. Viitattu 26.4.2022. DOI:10.48550/arxiv.1412.6980.

Kivinen, N., Koskela, A., Kauppinen, A. & Kaarniranta, K. 2017. Silmänpohjan ikärappeuman patogeneesi-autofagian ja inflammasomien vuoropuhelua. Lääketieteellinen aikakauskirja Duodecim. Helsinki: Suomalainen Lääkäriseura Duodecim. Vol. 133, No. 7, 641–647. Viitattu 25.1.2022.
<https://www.duodecimlehti.fi/duo13646>.

Kivinen, N. & Kaarniranta, K. 2021. Lasiaisinjektiot mullistivat silmänpohjan kostean ikärappeuman hoidon. Suomen Lääkärilehti. Helsinki: Suomen Lääkäriliitto. Vol. 77, 2055–2058. ISSN 0039-5560 (painettu). ISSN 2489-7434 (sähköinen). Viitattu 26.1.2022.
<https://www.laakarilehti.fi/tieteessa/katsausartikkeli/lasiaisinjektiot-mullistivat-silmanpohjan-kostean-ikarappeuman-hoidon/?public=2e4a8c085da88f06ec826c24daec1a5b>.

Kivinen, N., Tuulonen, A. & Kaarniranta, K. 2018. Silmänpohjan kuvantaminen yleistyy ja tarkentuu. Lääketieteellinen aikakauskirja Duodecim. Helsinki: Suomalainen Lääkäriseura Duodecim. Vol. 134, No. 12, 1235–1243. Viitattu 2.2.2022. <https://www.duodecimlehti.fi/duo14388>.

Kostea silmänpohjan ikärappeuma (AMD). Käypä hoito -suositus. 2016. Suomalaisen Lääkäriseuran Duodecimin, Suomen Silmälääkäriyhdistyksen ja sairaanhoitopiirien silmätautien klinikoiden asettama työryhmä. Helsinki: Suomalainen Lääkäriseura Duodecim. Viitattu 26.1.2022.
<https://www.kaypahoito.fi/hoi50107>.

Krenker, A., Bešter, J. & Kos, A. 2011. Introduction to the Artificial Neural Networks. Teoksessa: K. Suzuki (toim.). Artificial Neural Networks – Methodological Advances and Biomedical Applications. London, England: IntechOpen. 362 s. ISBN: 978-953-307-243-2. Viitattu 11.4.2022. DOI:10.5772/15751.

Kuoppala, J. & Härkönen, U. 2015. BIOLOGISET LÄÄKKEET SILMÄNPOHJAN KOSTEAN IKÄRAPPEUMAN HOIDOSSA. Loppuraportti. Lääkealan turvallisuus- ja kehittämiskeskus Fimea. 40 s. Viitattu 8.2.2022. <https://docplayer.fi/17616651-Biologiset-laakkeet-silmanpohjan-kostean-ikarappeuman-hoidossa.html>.

Li, J.Q., Welchowski, T., Schmid, M., Mauschitz, M.M., Holz, F.G. & Finger, R.P. 2020. Prevalence and incidence of age-related macular degeneration in Europe: a systematic review and meta-analysis. British Journal of Ophthalmology. London, United Kingdom: BMJ Publishing Group Ltd. Vol. 104, No. 8, 1077–1084. Viitattu 2.4.2022. DOI:10.1136/BJOPHTHALMOL-2019-314422.

Loukovaara, S. & Haukka, J. 2021. Silmätoimenpiteet Suomessa vuosina 2010–2016. Suomen Lääkärilehti. ISSN 0039-5560 (painettu). ISSN 2489-7434 (sähköinen). Helsinki: Suomen Lääkäriliitto. Vol. 77, 2487–2496. Viitattu 3.4.2022. <https://www.laakarilehti.fi/tieteessa/alkuperaistutkimukset/silmatoimenpiteet-suomessa-vuosina-2010-ndash-2016/>.

Mcculloch, W.S. & Pitts, W. 1943. A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY. BULLETIN OF MATHEMATICAL BIOPHYSICS. Vol 5, 115–133. Viitattu 7.4.2022. <https://home.csulb.edu/~cwallis/382/readings/482/mccolloch.logical.calculus.ideas.1943.pdf>.

Mckinney, W. 2010. Data Structures for Statistical Computing in Python. Viitattu 26.4.2022. <https://pandas.pydata.org>.

Mohri, M., Rostamizadeh, A. & Talwalkar, A. 2018. Foundations of Machine Learning. 2nd ed. F. Bach.(toim.). Cambridge, Massachusetts, USA: The MIT Press. 504 s. ISBN: 978-0-262-03940-6. Viitattu 4.4.2022.

<https://cs.nyu.edu/~mohri/mlbook/>.

Mustajoki, P. 2021. Lihavuus. Lääkärikirja Duodecim. Duodecim Terveyskirjasto. Helsinki: Kustannus Oy Duodecim. Viitattu 30.3.2022.

<https://www.terveyskirjasto.fi/dlk00042>.

Nguyen, T.P., Yeh, C.T., Cho, M.Y., Chang, C.L. & Chen, M.J. 2022. Convolutional neural network bidirectional long short-term memory to online classify the distribution insulator leakage currents. Electric Power Systems Research. Elsevier. Vol. 208, Article 107923. 11 s. Viitattu 11.4.2022.

DOI:10.1016/J.EPSR.2022.107923.

Nielsen, M.A. 2015. Neural Networks and Deep Learning. Determination Press. Viitattu 7.4.2022. <http://neuralnetworksanddeeplearning.com/chap1.html>.

Nielson, M. 2022. Neural Networks and Deep Learning. Y Combinator Research, nyk. OpenResearch. Viitattu 20.4.2022.

[https://eng.libretexts.org/Bookshelves/Computer_Science/Applied_Programming/Book%3A_Neural_Networks_and_Deep_Learning_\(Nielsen\)](https://eng.libretexts.org/Bookshelves/Computer_Science/Applied_Programming/Book%3A_Neural_Networks_and_Deep_Learning_(Nielsen)).

Nigri, A., Levantesi, S, Marino, M., Scognamiglio, S. & Perla, F. 2019. A Deep Learning Integrated Lee–Carter Model. Risks. Basel, Switzerland: MDPI. Vol. 7, No. 1. Article 33, 16 s. Viitattu 14.4.2022. DOI:10.3390/RISKS7010033.

Ojamo, M. & Tolkkinen, L. 2020. Näkövammarekisterin vuosikirja 2020. 56 s. ISSN:1236-5114. Viitattu 1.4.2022. <https://cms.nkl.fi/sites/default/files/2021-09/N%C3%A4k%C3%B6vammarekisterin%20vuosikirja%202020.pdf>.

Olah, C. 2015. Understanding LSTM Networks. Viitattu 21.4.2022.

<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.

Pedregosa, F., 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., Perrot, M. & Duchesnay, É. 2011. Scikit-learn: Machine Learning in Python. Teoksessa: M. Braun (toim.). *Journal of Machine Learning Research*. Vol. 12, No. 85, 2825–2830. Viitattu 10.5.2022.

<https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>.

Raftery, J., Clegg, A., Jones, J., Tan, C. & Lotery, A. 2007. Ranibizumab (Lucentis) versus bevacizumab (Avastin): modelling cost effectiveness. *British Journal of Ophthalmology*. London, United Kingdom: BMJ Publishing Group. Vol. 91, 1244–1246. Viitattu 3.4.2022. DOI:10.1136/bjo.2007.116616.

Raschka, S. 2018. MLxtend: Providing machine learning and data science utilities and extensions to Python’s scientific computing stack. *Journal of Open Source Software*. Vol. 3, No. 24, s. 638. Viitattu 26.4.2022.

DOI:10.21105/JOSS.00638.

Ravichandiran, S. 2018. *Hands-On Reinforcement Learning with Python: Master Reinforcement and Deep Reinforcement Learning Using OpenAI Gym and TensorFlow*. Birmingham, United Kingdom: Packt Publishing Ltd. 318 s. ISBN 978-1-78883-652-4.

Rein, D.B., Wittenborn, J.S., Zhang, X., Honeycutt, A.A., Lesesne, S.B. & Saaddine, J. 2009. Forecasting Age-Related Macular Degeneration Through the Year 2050: The Potential Impact of New Treatments. *Arch Ophthalmol*. Chicago, Illinois, USA: American Medical Association. Vol. 127, No. 4, 533–540. Viitattu 3.4.2022. DOI:10.1001/archophthalmol.2009.58.

Rosenblatt, F. 1958 THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN.

Psychological Review. Vol. 65, No. 6, 386–408. Viitattu 7.4.2022. DOI: 10.1037/h0042519.

Ruder, S. 2016. An overview of gradient descent optimization algorithms. *Insight Centre for Data Analytics: NUI Galway*. Dublin: Aylie Ltd. Viitattu 27.4.2022. DOI:10.48550/arXiv.1609.04747.

Sammut, C. & Webb, G.I. 2010. Mean Squared Error. Teoksessa: C. Sammut & G.I. Webb (toim.). Encyclopedia of Machine Learning. Boston, Massachusetts, USA: Springer US. s. 653. Viitattu 26.4.2022. DOI:10.1007/978-0-387-30164-8_528.

Sarker, I.H. 2021. Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science. Springer Nature Singapore Pte Ltd. Vol. 2, Article 160, 21 s. Viitattu 6.4.2022. DOI:10.1007/s42979-021-00592-x.

Schuster, M. & Paliwal, K.K. 1997. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing. Vol. 45, No. 11, 2673–2681. Viitattu DOI:10.1109/78.650093.

Seitsonen, S., Onkamo, P., Immonen, I. & Järvelä, I. 2009. Silmänpohjan ikärappeuman alttiusgeenien tunnistaminen - molekyyli­genetiikan menestystarina. Lääketieteellinen aikakauskirja Duodecim. Helsinki: Suomalainen Lääkäriseura Duodecim. Vol. 125, No. 21, 2360–2364. Viitattu 8.2.2022. <https://www.duodecimlehti.fi/duo98402>.

Seppänen, M. 2021. Silmänpohjan ikärappeuma (makuladegeneraatio). Lääkärikirja Duodecim. Duodecim Terveyskirjasto. Helsinki: Kustannus Oy Duodecim. Viitattu 26.1.2022. <https://www.terveyskirjasto.fi/dlk00922>.

Shalev-Shwartz, S. & Ben-David, S. 2014. Understanding Machine Learning: From Theory to Algorithms. 1st ed. Cambridge, United Kingdom: Cambridge University Press. 410 s. ISBN 978-1-107-05713-5. Viitattu 4.4.2022. <https://www.cs.huji.ac.il/w~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf>.

Sharma, S. 2017. What the Hell is Perceptron?: The Fundamentals of Neural Networks. Towards Data Science. Viitattu 7.4.2022. <https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53>.

Sipilä, R., Tuulonen, A. & Kaarniranta, K. 2018a. VEGF:n estäjähoidon valinta AMD:ssä. Vältä viisaasti. Helsinki: Suomalainen Lääkäriseura Duodecim. Viitattu 3.4.2022. <https://www.kaypahoito.fi/dnd00061>.

Sipilä, R., Tuulonen, A. & Kaarniranta, K. 2018b. VEGF:n estäjähoidosta luopuminen AMD:n hoidossa. Vältä viisaasti. Helsinki: Suomalainen Lääkäriseura Duodecim. Viitattu 3.4.2022. <https://www.kaypahoito.fi/dnd00060>.

Sutton, R.S. & Barto, A.G. 2020. Reinforcement learning: An introduction. 2nd ed. F. Bach. (toim.). Cambridge, Massachusetts, USA: The MIT Press. 552 s. (Adaptive Computation and Machine Learning). ISBN:9780262039246. Viitattu 7.4.2022. <http://incompleteideas.net/book/RLbook2020.pdf>.

Wong, W.L., Su, X., Li, X., Cheung, C.M.G., Klein, R., Cheng, C.Y. & Wong, T.Y. 2014. Global prevalence of age-related macular degeneration and disease burden projection for 2020 and 2040: A systematic review and meta-analysis. *The Lancet Global Health*. Elsevier Ltd. Vol. 2, No. 2, e106–e116. Viitattu 1.4.2022. DOI:10.1016/S2214-109X(13)70145-1.

Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klinger, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M. & Dean, J., 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. Viitattu 11.4.2022. DOI:10.48550/arxiv.1609.08144.

Zhang, R.-H., Liu, Y.-M., Dong, L., Li, H.-Y., Li, Y.-F., Zhou, W.-D., Wu, H.-T., Wang, Y.-X. & Wei, W.-B. 2022. Prevalence, Years Lived With Disability, and Time Trends for 16 Causes of Blindness and Vision Impairment: Findings Highlight Retinopathy of Prematurity. Teoksessa: A. Chu (toim.). *New Frontiers in Pediatrics*. Frontiers Media S.A. Vol. 10, Article 735335. 9 s. Viitattu 1.4.2022. DOI:10.3389/FPED.2022.735335.