



TEKNILLINEN TIEDEKUNTA

KEINOTEKOISET NEUROVERKOT LUOKITTELUONGELMISSA

Joni Salonen

PROSESSITEKNIIKAN TUTKINTO-OHJELMA

Kandidaatintyö

Maaliskuu 2021

TIIVISTELMÄ

Keinotekoiset neuroverkot luokitteluongelmissa

Joni Salonen

Oulun yliopisto, Prosessitekniikan tutkinto-ohjelma

Kandidaatintyö 2021, 39 s.

Työn ohjaajat yliopistolla: TkT Aki Sorsa sekä DI Antti Koistinen

Tämä kandidaatintyö koostuu teoriaosasta sekä kokeellisesta osiosta. Työn koostuessa sekä teoriaosasta että kokeellisesta osiosta, lukijalle tarjotaan teoriaosiossa pintaraapaisu tiedonlouhinnan sekä keinotekoisien neuroverkkojen pariin. Kerrottaessa tiedonlouhinnasta esitellään tiedonlouhinnan merkitystä nykypäivän yhteiskunnalle, jossa kerättävän ja varastoitavan datan määrä kasvaa lähes eksponentiaalisella vauhdilla. Tiedonlouhinnan yhteydessä esitellään kolme tyypillistä tiedonlouhinnan tuottamaa mallia sekä esitellään tiedonlouhinnassa käytettävä CRISP-DM-toimintamalli, jonka avulla tiedonlouhintaprosessit saavat kronologisesti etenevät suuntaviivat. Tiedonlouhinnan lisäksi työssä esitellään keinotekoisien neuroverkkojen keskeisimpiä periaatteita perehtymällä siihen, miten keinotekoiset neuroverkot todellisuudessa rakentuvat, oppivat ja vaikuttavat eri sovellusten kautta ihmisten elämään. Keinotekoisien neuroverkkojen luokittelukykyä havainnollistetaan työn kokeellisessa osiossa, jota varten työhön on etsitty luokittelua vaativa aineisto. Kokeellisessa osiossa esitellään yksi mahdollisista neuroverkkoluokittimen muodostamistavoista hyödyntäen numeerisen laskennan tietokoneohjelmisto MATLAB[®]:ia. MATLAB[®] tarjoaa suoraviivaisen menettelytavan keinotekoisien neuroverkkojen opettamiseen tietyn luokitteluongelman ratkaisemiseksi. Kokeellisen osion lopussa neuroverkkoluokittimen avulla saatua lopputulosta arvioidaan, jonka perusteella laaditaan yhteenveto keinotekoisien neuroverkkojen käytettävyydestä erilaisten luokitteluongelmien yhteydessä.

Asiasanat: CRISP-DM, keinotekoiset neuroverkot, luokittelu, tiedonlouhinta

SISÄLLYSLUETTELO

1 Johdanto	6
2 Tiedonlouhinta	8
2.1 Johdanto tiedonlouhintaan	8
2.2 Tiedonlouhinnan malleja.....	8
2.2.1 Luokittelu	9
2.2.2 Regressio	9
2.2.3 Klusterointi.....	10
2.3 CRISP-DM.....	11
3 Keinotekoiset neuroverkot luokitteluongelmissa.....	14
3.1 Johdanto keinotekoisiiin neuroverkkoihin	14
3.2 Keinotekoisien neuroverkkojen rakenne.....	16
3.3 Neuronit.....	17
3.3.1 Painokerroin	17
3.3.2 Bias.....	17
3.3.3 Aktivaatiofunktio	18
3.4 Sigmoid-funktio	20
3.5 Keinotekoisien neuroverkkojen opettaminen.....	21
3.5.1 Gradienttimenetelmä	22
3.5.2 Backpropagation-algoritmi	23
3.6 Keinotekoisia neuroverkkoja hyödyntäviä luokittelutilanteita	24
4 Neuroverkkoluokittimen muodostaminen.....	26
4.1 Kokeellisen osion tavoite ja valitun tietokannan esittely.....	26
4.2 Tietokannan esikäsittely.....	26
4.3 Mallin muodostaminen	31
5 Saatujen tulosten arviointi.....	36
6 Yhteenveto	38
Lähdeluettelo	

MERKINNÄT JA LYHENTEET

ANN	Artificial Neural Network, suom. keinotekoinen neuroverkko
CRISP-DM	Cross Industry Standard Process for Data Mining
IoT	Internet of Things, suom. esineiden internet
Mnist	Modified National Institute of Standards and Technology database
ReLU	Rectified Linear Unit
E	Virhefunktion arvo
a	Painokertoimen summattu arvo, aktivaatio
a_j^l	Kerroksen l neuronin j aktivaation suuruus
b	Bias, vakiotermi
b_j^l	Kerroksen l neuronin j vakiotermin suuruus
E	Neperin luku, luonnollisten logaritmien kantaluku
M	Neuronien lukumäärä virhefunktiossa
n	Lukumäärä
t_k	Neuroverkon syötteelle k antama tulos
w_i	Painokertoimen i arvo
w_n	Painokertoimien lukumäärä
w_{ij}^l	Kerroksen l-1 neuronin i ja kerroksen l neuronin j välinen painokerroin
x_i	Neuronin i käyttämän yhteyden järjestysnumero
x_n	Neuroniin tulevien tulosignaalien lukumäärä
y_k	Neuroverkon tuottaman tuloksen tavoitearvo
z_j^l	Kerroksen l neuronin j summa
ρ	Määrää Sigmoid-funktion muodon, suuret arvot tekevät käyrästä tasaisemman ja pienet jyrkemmän. Usein annetaan implisiittisesti arvoksi 1.
φ	Aktivaatiofunktio
Abs	Komento, joka palauttaa muuttujan r absoluuttisten elementtien arvot

Categorical	Komento, jolla saadaan määriteltyä lähtömuuttujat luokkiin 0–9
Corr	Komento, jolla viitataan korrelaatioon
Dummyvar	Komento, joka muuntaa lähtömuuttujat dummy-muuttujiksi niin, että yksi sarake sisältää kaikki 10 muuttujaa, jotka saavat vain joko arvoja 0 tai 1.
Ind1	Määrittelee tässä työssä muuttujat, joiden keskihajonta on 0
Ind2	Määrittelee tässä työssä muuttujat, joiden korrelaation itseisarvo on < 0.2
Input	Opetusaineiston tulomuuttujat
Inputtest	Testausaineiston tulomuuttujat
Mnisttest	Testausaineiston sisältävä Mnist-tietokanta
Mnisttrain	Opetusaineiston sisältävä Mnist-tietokanta
Output	Opetusaineiston lähtömuuttujat
Outputtest	Testausaineiston lähtömuuttujat
Plot	Piirtokomento
Std	Komento, joka laskee tässä työssä tulomuuttujien seasta arvot, joiden keskihajonta on 0.
Xlabel	Komento, jolla saadaan vaaka-akselille otsikko
Ylabel	Komento, jolla saadaan pystyakselille otsikko
r	Muuttuja, jolla kuvataan tulo- ja lähtömuuttujien välistä korrelaatiota
[]	Kertoo tulomuuttujan olevan määrittelemätön, jolloin tulomuuttujaa voidaan määrittää jälkeen päin.
;	Merkitään yleisesti tietyn koodin perään. Estää välivaiheiden/ratkaisun esiintulon.
(:,n)	Komento, jolla saadaan määrättyä matriisista tietty rivi.

1 JOHDANTO

Digitalisaatio ja tietotekniikan kehittyminen vuosien saatossa ovat vaikuttaneet merkittävästi siihen, kuinka voimme 2020-luvulla analysoida, hyödyntää ja kerätä erilaista dataa. Erityisesti tietotekniikan kehittymisen myötä suurien datamäärien kerääminen ja säilyttäminen on tehty nykyteknologian avulla helpoksi. Dataa kerääviä tahoja on monipuolisesti eri aloilla aina päivittäistavarakaupoista raskaaseen prosessiteollisuuteen, koska kerätty data luo nykypäivänä suurta lisäarvoa sitä keränneelle toimijalle. Kerätyn datan tuoma lisäarvo piilee itse datassa, koska analysoimalla ja käsittelemällä saatua dataa oikein saattaa se paljastaa sellaista hyödyllistä informaatiota, jolla voidaan pyrkiä saamaan etumatkaa kilpaileviin yrityksiin. On siis hyvä ymmärtää se ero, joka datalla ja tiedolla on. Data itsessään ei omaa juurikaan painoarvoa, mutta tiedonlouhinnan avulla datasta saadaan mahdollisesti johdettua informaatiota eli tietoa, jolla on jokin syvempi merkitys.

Tiedonlouhinta on termi, joka vaikuttaa keskeisesti datan analysointiin ja käsittelyyn. Eri toimijat keräävät valtavia määriä dataa, joten on tärkeää osata löytää suurten datamassojen seasta esimerkiksi tuotekehityksen kannalta merkityksellinen aines. Data itsessään ei luo lisäarvoa tuotekehityksen kannalta, jos sille suoritettava tiedonlouhinta on puutteellista. Kerätyn datan määrä ja laatu vaikuttavat olennaisesti tiedonlouhinnan avulla saavutettavaan lopputulokseen. Datasta on pystyttävä erottamaan hyödytön ja virheellinen aines. Huolellisella esityöllä tiedonlouhinnan onnistumisen mahdollisuutta saadaan tehostettua, kun taas hätiköidyllä esityöllä dataan jää helposti hyödytöntä ja virheellistä ainesta, jotka johtavat väistämättä hyödyttömään ja virheelliseen lopputulokseen. Tiedonlouhinnan helpottamiseksi on kehitetty erinäisiä menetelmiä, joilla tiedonlouhintaa voidaan harjoittaa. Tässä työssä dataa lähestytään käyttämällä apuna CRISP-DM -toimintamallia.

Neurotieteen tutkimuskohteita ovat muun muassa hermoston toiminta ja sen rakenne sekä neurobiologia, jonka tutkimusalaa ovat neuronit eli hermosolut. Ihmisaivot älykkyyksineen ovat monen tekijän summa, mutta ihmisälykkyyteen viitatessa hermosoluilla on merkittävä rooli tiedonvälittäjänä toimiessaan. Tätä älykkyyttä pyritään mallintamaan parhaan mukaan esimerkiksi erilaisten itsenäisten koneiden ja robottien käyttöön, jolloin ne pystyisivät suorittamaan älyllisiä toimintoja vaativia tehtäviä. Tätä

älyllisen toiminnan matkimista pyritään kehittämään keinotekoisten neuroverkkojen avulla. Keinotekoisten neuroverkkojen avulla pyritään mallintamaan tapaa, joilla ihmisaivot suorittavat jonkin tietyn mielenkiinnon kohteena olevan älyllisen tehtävän tai toiminnon (Haykin, 2009, s. 2).

Keinotekoisten neuroverkkojen käyttö perustuu niiden opettamiseen. Niitä voidaan opettaa esimerkiksi tiettyjen datakokoelmien avulla tiettyjen luokitteluongelmien tunnistamiseksi. Erilaiset kuvantunnistusta vaativat luokitteluongelmat ovat erinomaisia esimerkkejä luokitteluongelmista, joihin keinotekoiset neuroverkot soveltuvat. Keinotekoisii neuroverkkoihin pohjautuvia sovellutuksia on monilla eri aloilla aina lääketieteestä tietotekniikan kautta prosessiteollisuuteen.

Tiivistelmässä esitettiin työssä toistuvat avainsanat. Työn seuraamisen helpottamiseksi on syytä avata työn keskeisten käsitteiden sisältö:

- **CRISP-DM** on data-analytiikassa käytettävä toimintamalli data-aineiston lähestymiseen, mikä ei ole riippuvainen datan tyypistä tai käytettävistä työkaluista (North, 2012, s. 5).
- **Keinotekoiset neuroverkot** ovat massiivisia hajautettuja prosessoreita, jotka koostuvat yksinkertaisista prosessointiyksiköistä. Keinotekoisilla neuroverkoilla on luonnollinen taipumus oppia päätöksistään. (Haykin, 2009, s. 2)
- **Luokittelu** on yksi tiedonlouhinnan tuottamista malleista. Luokittelun tavoitteena on luoda ryhmittymiä tietojoukoista havaittujen attribuuttien samankaltaisuuden perusteella. (North, 2012, s. 238)
- **Tiedonlouhinta** on laskennallinen prosessi, jolla analysoidaan suurikokoisia tietojoukkoja tilastollisin ja loogisin menetelmin. Tiedonlouhinnan tavoitteena on paljastaa tuntemattomat ja mielenkiintoiset mallit, jotka voivat paljastaa hyödyllistä informaatiota. (North, 2012, s. 240)

2 TIEDONLOUHINTA

2.1 Johdanto tiedonlouhintaan

Tiedonlouhinnan merkitys on noussut yhä merkittävämpään asemaan vuosi vuodelta, sillä maailmassa olevan datan määrä kasvaa lähes eksponentiaalisella vauhdilla. Datan määrän nopeaa kasvua ei voida selittää yksiselitteisesti vaan kyseessä on lukuisten tekijöiden summa. Internetin leviäminen yhä useampien ihmisten käytettäväksi, yritystoiminta, yhteiskunta, erilaiset tieteen- ja tekniikan verkostot sekä viimeisimpänä IoT eli esineiden internet tuottavat uutta dataa tiedonlouhijoiden käsiteltäväksi enenevässä määrin. Kasuvat datamäärät omaavat sellaisen määrän potentiaalia ja mahdollisuuksia, että olisi typerää jättää hyödyntämättä niitä. Valtavat datamäärät eivät kuitenkaan tarjoa automaattisesti hyödyllistä informaatiota suurista potentiaaleistaan huolimatta vaan ne toimivat ikään kuin ajavana voimana sille, miksi tiedonlouhinnan merkitys on kasvanut niin voimakkaasti viimeisimpien vuosien aikana. Tiedonlouhinta on olennainen prosessi, jossa käytetään erilaisia menetelmiä laadukkaiden datamallien tuottamiseksi. Käytettävän tiedonlouhintamallin valinta on olennainen osa tiedonlouhinta. (Garcia et al., 2015, s. 1–2)

Tiedonlouhinnan voidaan katsoa olevan osana lähes jokaisen liiketoimintaa harjoittavan yrityksen toimenkuvaa, sillä nykypäivänä useimmat yritykset keräävät dataa päivittäisistä toimistaan. Useat prosessiteollisuuden yritykset operoivat esimerkiksi aktiivisesti viennin parissa. Asiakkaalta kerättävää dataa voi olla esimerkiksi tilauksien säännöllisyys sekä tilattavan tuotteen määrä ja laatuominaisuudet. Kerättyä dataa analysoimalla datasta on mahdollista muodostaa arvokasta informaatiota sisältäviä malleja, joita hyödyntämällä voidaan parhaimmillaan tuottaa lisäarvoa sekä yritykselle, että yrityksen kanssa vuorovaikutuksessa toimiville asiakkaille. Tiedonlouhinnan tuloksena saatavat mallit voidaan jakaa niiden luonteesta riippuen joko ennustaviin tai kuvaileviin malleihin. Tutkittavasta data-aineistosta riippuen on syytä miettiä, halutaanko lopputulemana saatava malli kuvata ennustuksiin perustuen vai kuvailemalla. (Garcia et al., 2015, s. 3).

2.2 Tiedonlouhinnan malleja

2.2.1 Luokittelu

Luokittelu on tiedonlouhintamalli, jossa on olemassa valmiiksi erilliset luokat, joihin jako voidaan suorittaa. Luokittelun tavoitteena on jakaa aineisto niin, että tiettyyn luokkaan kuuluvien kohteiden välillä olisi jokin yhtenäinen ominainen piirre. Päättöspuiden toiminta on luokitteluun perustuva tiedonlouhinnan tuottama malli, jonka avulla voidaan suorittaa esimerkiksi yksinkertainen binäärinen luokittelu. Esimerkiksi, jos eläin haukkuu, niin se on koira, kun taas vastaavasti, jos eläin naukuu, niin se on kissa. (Garcia et al., 2015, s. 5). Keinotekoisia neuroverkkoja hyödyntävä tyypillinen luokittelutapaus on esimerkiksi sähköpostin roskapostisuodattimen toiminnan tarkasteleminen. Roskapostisuodatin voidaan opettaa tunnistamaan saapuvien merkityksellisten sähköpostien seasta merkityksettömät roskapostit, jotka se kykenee luokittelemaan automaattisesti esimerkiksi tunnistamalla viestin seasta jonkinlaisen tietyn alatyylisen sanan tai mainoksen.

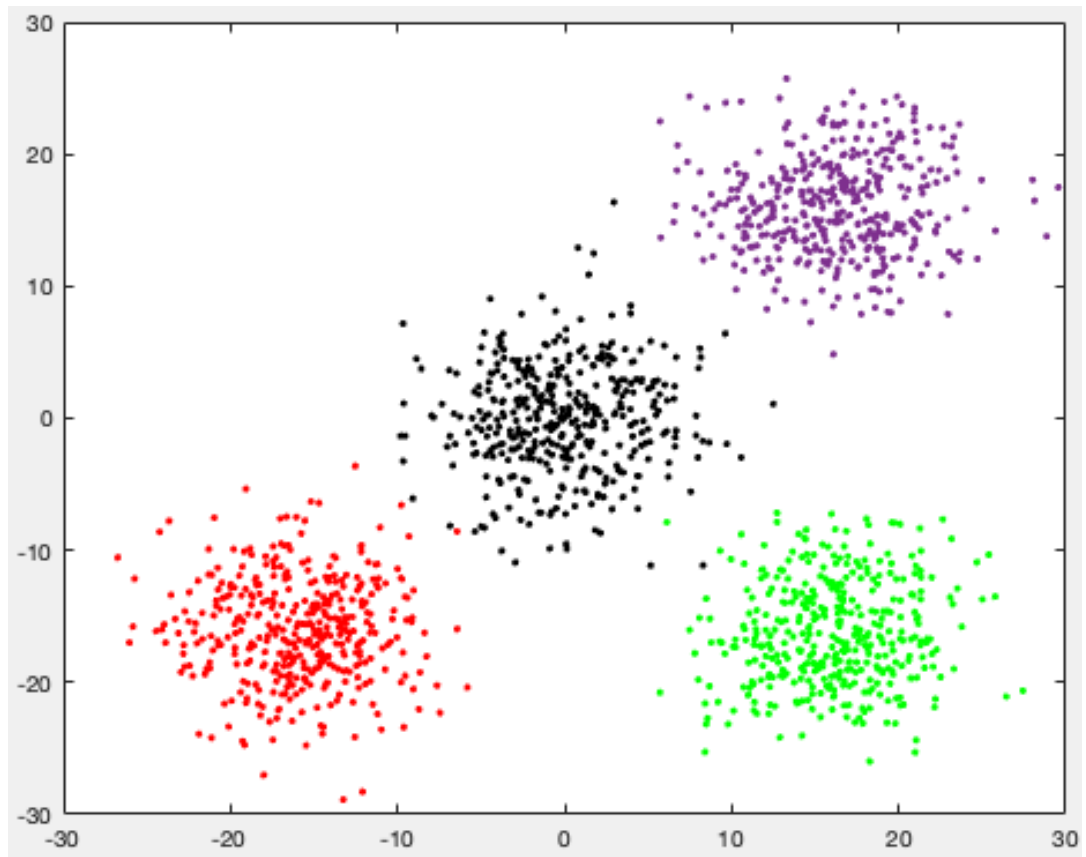
2.2.2 Regressio

Regressio on yleisesti tilastotieteen parissa paljon käytetty tutkimusmenetelmä sekä samalla yksi vanhimmista tiedonlouhinnan tuottamista malleista (Garcia et al., 2015, s. 3). Regression tarkoituksena on tutkia eri muuttujien välisiä vuorovaikutuksia toistensa suhteen. Regressio on erinomainen valinta tiedonlouhinnan malliksi esimerkiksi silloin, jos tehtävänä on tuottaa datasta informaatiota, jossa tarkastellaan eri muuttujien välisiä suhteita. Regressiota käytettäessä valitaan yleisesti jokin tietty selitettävä muuttuja, jonka vaikutuksesta ollaan erityisen kiinnostuneita. Datassa esiintyvät loput muuttujat eivät ole samanlaisessa keskiössä tarkastelun suhteen, sillä jäljellejääneiden muuttujien avulla pyritään kuvaamaan tai ennustamaan sen muuttujan käyttäytymistä, joka on valittu erityisen mielenkiinnon kohteeksi. Regressiota käytettäessä tiedonlouhinnan mallina on hyvä huomioida se, että regressio voidaan jakaa kahteen pääluokkaan eli lineaariseen- ja epälineaariseen regressioon, joiden lisäksi on olemassa erikoistapaukseksi luettava logistinen regressio. Lineaarissa regressiossa datan käyttäytymistä pyritään ennustamaan algebrallisen kaavan avulla, jossa tutkitaan miten muuttujista saadut havainnot sijoittuvat lineaarisen suoran läheisyyteen, kun taas esimerkiksi logistisessa regressiossa ennustetaan neliöllistä kaavaa apuna käyttäen sitä, mikä on havaintojen

lopullinen tulos ja mikä on todennäköisyys ennusteen olemiseksi todellinen tulos (North, 2012, 243–244). Regressiosta puhuttaessa puhutaan yleisesti termistä korrelaatio, jolla viitataan havaintojen samankaltaisuuteen. Korrelaation arvolla -1 muuttujilla on täydellinen negatiivinen korrelaatio, arvolla 0 korrelaatiota ei havaita ja arvolla 1 muuttujilla on keskenään täydellinen korrelaatio. (North, 2012, s. 239)

2.2.3 Klusterointi

Klusterointi on eräänlainen ohjaamattomaan oppimiseen kuuluva tapa luokitella data-aineistosta saatavia havaintoja, jota käytetään silloin, kun valmiita luokkakategorioita ei ole, mutta havainnot on pystyttävä jakamaan luonnollisiin ryhmiin eli klustereihin (Garcia et al., 2015, s. 5). Ohjaamattomalla oppimisella tarkoitetaan sellaista oppimisalgoritmia, jossa tietyt johtopäätökset tehdään käsiteltävän data-aineiston perusteella niin, että johtopäätösten teko perustuu ainoastaan käsiteltyyn dataan ilman tietoa todellisista vastauksista. Klusterointia käytetään data-analytiikassa silloin, kun data-aineistosta pyritään löytämään jonkinlaisia mahdollisia luokkia tai malleja ilman valmiita luokkakategorioita. Klusteroinnin tavoitteena on jakaa datasta saatava tieto sellaisiin luonnollisiin klustereihin, joissa jokaiseen itsenäiseen klusteriin kuuluvien havaintojen yhteneväisyys olisi mahdollisimman suurta. (Haykin, 2009, s. 242) Ryhmittymisen klustereihin voidaan siis katsoa tapahtuvan mittaamalla klusterien sisäistä samankaltaisuutta eli homogeenisuutta. Klusteroinnin tavoitteena on, että yksittäisessä klusterissa esiintyvien havaintopisteiden homogeenisuus olisi mahdollisimman korkea, mutta klusterit itsessään olisivat keskenään toisiinsa nähden mahdollisimman heterogeenisia. Klusterointia voidaan harjoittaa monen eri tavan kautta, mutta yksi yleisimmin käytetyistä klusterointimenetelmistä on k-means-klusterointimenetelmä. K-means muistuttaa jossain määrin perinteistä luokittelua sillä erolla, että datasta saatavia havaintoja varten ei ole annettuna valmiita kategorisoituja luokkia. (North, 2012, s. 101) Kuvassa 1 on esimerkki klusteroinnista, jossa klusterit ovat määrittäneet selkeästi neljään erilliseen klusteriin. Kuvasta 1 voidaan havaita klusterin sisällä vallitseva homogeenisuus ja klustereiden välillä vallitseva heterogeenisuus.



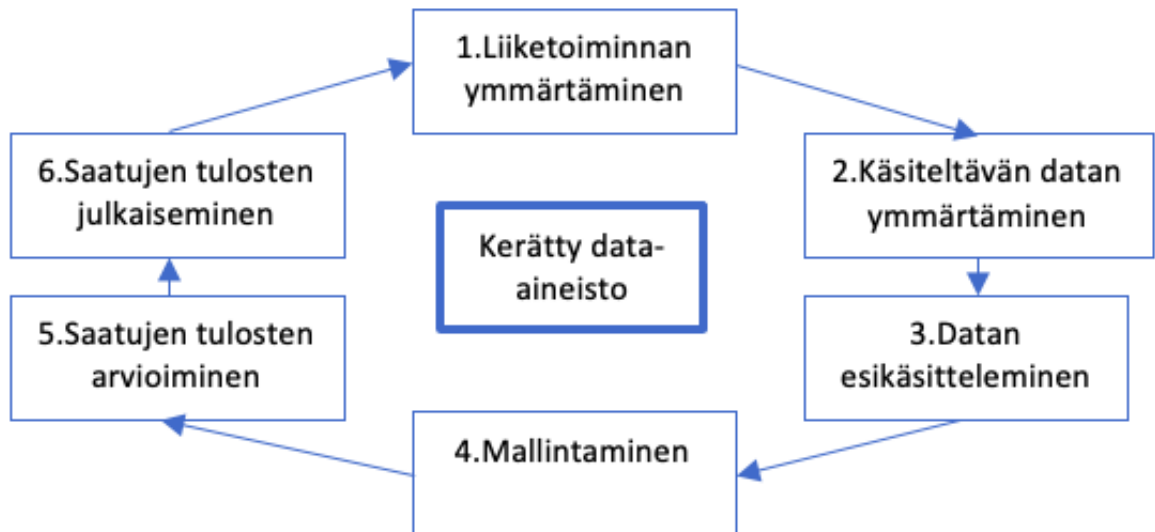
Kuva 1. Neljä homogeenisesta klusterista, jotka ovat keskenään heterogeenisiä.

2.3 CRISP-DM

CRISP-DM eli Cross Industry Standard Process for Data Mining on vapaasti saatavilla oleva, standardoitu toimintamalli tiedonlouhintaa vaativien projektien toteuttamiseksi. Vuonna 1996 kiinnostus tiedonlouhintaa kohtaan alkoi ottamaan ensimmäisiä enenevissä määrin merkittäviä askeleita, mutta mitään yleisesti hyväksytyjä lähestymistapoja tiedonlouhinnan suorittamiseksi ei ollut. Koettiin, että standardoitu teollisuudesta riippumaton tiedonlouhintamalli auttaisi organisaatioita hyödyntämään tiedonlouhinnan avulla saavutettavia tuloksia tehokkaammin, joka voisi heijastua organisaatioiden kehittymisen kautta talouteen positiivisessa mielessä. Tästä syystä kehitettiin data-analytiikkaan soveltuva toimintamalli, jonka tarkastelutapa perustuu siihen, että yhden suuren projektin sijaan on mielekkäämpää ajatella samaisen projektin koostuvan enneminkin kuudesta pienemmästä välietapistä. Purkamalla yksi suuri projekti pienemmiksi välietapeiksi autetaan yrityksiä ymmärtämään paremmin, mitkä kaikki tekijät on otettava huomioon tiedonlouhinnan eri vaiheissa sekä mitkä kaikki tekijät

vaikuttavat tiedonlouhintaprosessin onnistumiseen. Seuraamalla CRISP-DM:n sisältäviä välivaiheita yritykset saavat selkeät suuntaviivat tiedonlouhintaprojektin suorittamiseksi, jolloin tiedonlouhinnan onnistumismahdollisuudet kasvavat. CRISP-DM:n suosio data-analytiikassa perustuu siihen, että se on tarpeeksi kattava toimintamalli tarjoamaan yksinkertaisen tavan täydellisesti suoritettavan tiedonlouhintahankkeen suorittamiseksi (Shearer, 2000, s. 13–14). Toimintamallin sisältämät kuusi välietappia ja niiden keskeisimmät tavoitteet ovat:

1. Liiketoiminnan ymmärtäminen. Liiketoiminnan ymmärtämisen keskeisin tavoite on luoda kuva liiketoiminnan tavoitteista, arvioida vallitseva tilanne, määrittää tiedonlouhinnan tavoitteet sekä laatia projektisuunnitelma tavoitteeseen pääsemiseksi.
2. Datan ymmärtäminen. Datan ymmärtämisen keskeisin tavoite on kerätä tarvittavat lähtötiedot, kuvailla ja tutkia kerättyä dataa sekä varmistaa, että data on laadullisesti ja määrällisesti riittävä tiedonlouhinnan suorittamiseen.
3. Datan valmisteleminen. Datan valmistelemisen keskeisin tavoite on rajata tiedonlouhintaan päätyvä lopullinen aineisto sekä karsia epäolennainen data olennaisen seasta.
4. Mallintaminen. Mallintamisen keskeisin tavoite on valita mallintamiseen käytettävä tiedonlouhintamalli, tuottaa mallista prototyyppi, rakentaa varsinainen tiedonlouhintaa kuvaava malli ja arvioida rakennettua mallia.
5. Arviointi. Arvioinnin keskeisin tavoite on arvioida saatuja lopputulemia sekä määrittää mahdolliset seuraavat toimenpiteet.
6. Tulosten julkaiseminen. Tulosten julkaisemisen keskeisin tavoite on suunnitella käyttöönotto, tuottaa varsinainen kuvaus suoritetusta tiedonlouhinnasta ja arvioida, vastasiko lopputulema liiketoiminnan ymmärtämisessä määritettyä tiedonlouhinnan tavoitetta.



Kuva 2. CRISP-DM:in välivaiheita havainnollistava kuvaaja. (mukaiillen, North, 2012, s. 6)

Kuvassa 2 on esitetty CRISP-DM -toimintamallia havainnollistava kuva. Siitä voidaan huomata, että kerätty data-aineisto on CRISP-DM -toimintamallia käytettäessä työn kannalta keskeisessä asemassa. Minkä tahansa tiedonlouhintaprojektin onnistumisen kannalta on tärkeää, että liiketoiminnan tavoite on selkeä ja ymmärrettävä, mitä kaikkia työvaiheita tavoitteeseen pääsemiseen edellyttää. Ihmisillä on taipumus aliarvioida liiketoiminnan ymmärtämisen merkitystä, koska se koetaan aikaa vievänä ja turhana välivaiheena. Työvaiheen vähättelyä perustellaan sillä, että ihmisillä on luontainen taipumus pyrkiä tuottamaan mielenkiintoisia vastauksia kysymyksiin, joihin haetaan vastauksia. (North, 2012, s. 6) Huolimatta CRISP-DM:n arvostuksesta ja levinneisyydestä eri aloille, sitä ei ole tarkoituksena pitää minään ohjekirjana, joka luo yritykselle välittömästi avaimen onneen. CRISP-DM on yhdessä erilaisten tiedonlouhintamallien ja ammattimaisten tiedonlouhijoiden esittämänä kuitenkin äärimmäisen arvokas työkalu auttamaan maltillisemmin tiedonlouhintaa harrastaneita henkilöitä, jolloin he voivat oivaltaa tiedonlouhintaprojektien ja siihen liittyvien erilaisten välivaiheiden merkityksen syvällisemmin. (Shearer, 2000, s. 19)

3 KEINOTEKOISET NEUROVERKOT LUOKITTELUONGELMISSA

3.1 Johdanto keinotekoiisiin neuroverkkoihin

Keinotekoisten neuroverkkojen rakenne, toiminta sekä tekoäly yleisestikin mielletään usein ihmisten keskuudessa hyvin korkealentoisiksi. Vaikutelmaan vaikuttaa moni tekijä, kuten esimerkiksi se, että aiheet ovat sellaisia, joita ei juurikaan sivuta esimerkiksi opetussuunnitelmissa. Media luo tiettyä kuvaa keinotekoisista neuroverkoista ja tekoälystä uutisoiden mitä teknisimpiä ja monimutkaisimpia sovelluskohteita, joihin neuroverkoilla ja tekoälyllä pystytään vaikuttamaan. Hakusanan tekoäly tuottamien kuvahaun tuloksien seassa on mitä utopistisimpia kuvia, joissa ihmishahmon pään sisään kuvataan mitä moniulotteisimpia kykyjä eri askareiden suorittamiseksi. Nämä tekijät ovat esimerkkejä, jotka luovat keinotekoisista neuroverkoista ja tekoälystä vaikutelman, jonka mukaan niiden rakenteen ja toiminnan ymmärtäminen olisi mahdollista vain lahjakkaimmille henkilöille. On totta, että syvällisemmän ymmärtämisen luominen kyseisistä käsitteistä vaatii aikaa ja aiheeseen paneutumista, mutta esimerkiksi keinotekoisten neuroverkkojen perusteiden ymmärtäminen on suhteellisen yksinkertaisesti havainnollistettavissa jo lyhyemmälläkin perehtymisellä aiheeseen.

Puhuttaessa keinotekoisista neuroverkoista liitetään asiayhteyteen usein termi black box. Black box -mallilla tarkoitetaan keinotekoisten neuroverkkojen yhteydessä sitä, että neuroverkkoon syötettävän data-aineiston ja verkon tuottama tulos ovat selkeästi havaittavissa, mutta välivaiheet data-aineiston syöttämisestä neuroverkon tuottamaan lopputulemaan eivät ole selkeästi tulkittavissa. Ilman konkretiaa on vaikea ymmärtää sitä, mitä välivaiheita laatikon sisällä tapahtui. Miten ne tapahtuivat? Miksi ne tapahtuivat? (Starmer, 2020) Black box -malli ei rajoitu pelkästään keinotekoiisiin neuroverkkoihin, vaan termiä voidaan käyttää myös muissa yhteyksissä, kuten prosessiteollisuudessa. Prosessien dynamiikkaa tarkastellessa voidaan törmätä esimerkiksi tilanteeseen, jossa tiedetään reaktoriin menevän syötteen koostumus, mutta ei tiedetä sitä, että miksi lähtevän syötteen koostumus eroaa merkittävästi arvioidusta lähtevästä koostumuksesta.

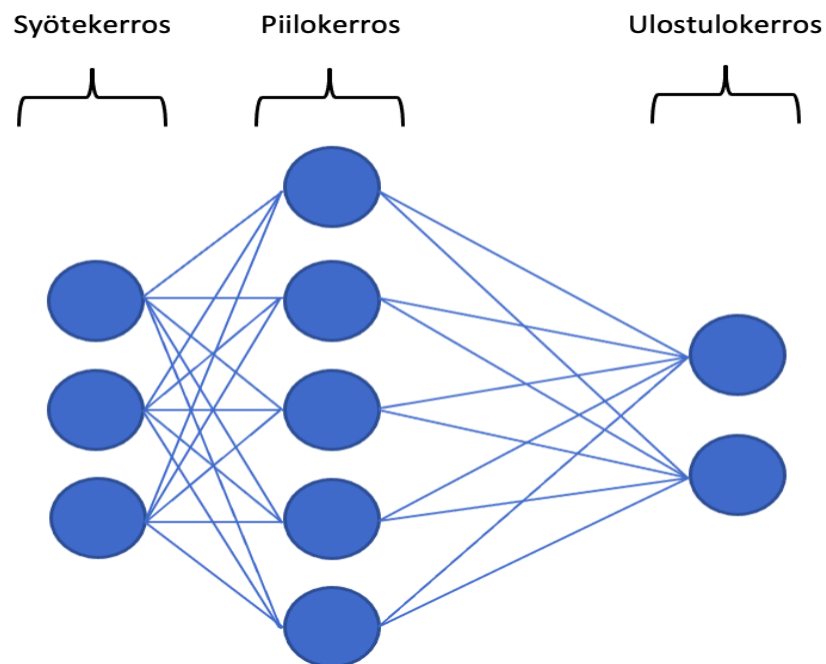
Keinotekoiset neuroverkot itsessään ovat yksi käytetyimmistä algoritmeista, jotka luovat pohjan koneoppimisella ja syväoppimisella saavutettaville mahdollisuuksille. Keinotekoiset neuroverkot ovat yksinkertaisista prosessointisoluista eli neuroneista koostuvia verkostoja. Niiden prosessointikyky perustuu neuronien välisien yhteyksien painotuksiin, jotka ovat peräisin tutkitusta prosessista. Prosessointikykyä pyritään matkimaan neurotieteen näkökulmasta, jonka tutkimuskohteita ovat ihmisen hermoston toiminta ja sen rakenne. Ihmisaivoissa on arvioitu olevan keskimäärin noin 100 miljardia hermosolua, jotka välittävät tietoa sähköisten impulssien avulla seuraaville hermosoluille. Sähköiset impulssit liikkuvat sähkökemiallisesti hermosolujen välillä olevien yhteyksien eli synapsien välillä. Hermosoluissa on tyypillisesti useita tulosignaaleja, jotka muuttuvat hermosolussa lähtösignaaleiksi. Yksittäinen lähtösignaali voi olla kuitenkin tulosignaali tuhansille muille hermosoluille, jonka vuoksi neuroverkot pystyvät luokittelemaan tietoa tehokkaasti. (Gurney, 2004, s. 13–14) Tätä ihmisaivojen prosessointikykyä keinotekoisilla neuroverkoilla pyritään matkimaan.

Koneoppimisella tarkoitetaan tekoälyn osa-aluetta, jossa kone oppii tarkasteltavan järjestelmän lainalaisuudet itsenäisesti. Usein koneoppimisessa käytetään useista kerroksista koostuvia keinotekoisia neuroverkkoja, jotka oppivat niille syötetyn data-aineiston pohjalta. (Tuominen, 2017) Erilaiset luokitteluongelmat ovat esimerkiksi oiva esimerkki siitä, kuinka keinotekoisia neuroverkkoja opettamalla ne pystyvät tuottamaan yhä laadukkaampia ja yhä todellisuudenmukaisempia lopputulemia. Keinotekoisien neuroverkkojen kyky oppia perustuu siihen, että niiden analysoiman data-aineiston perusteella analysoidusta lopputulemasta annetaan palautetta sen suhteen, kuinka kvalitatiivisesti sekä kvantitatiivisesti keinotekoinen neuroverkko on esimerkiksi onnistunut automaattisesti luokittelemaan sähköpostiin tulleet roskapostiviestit tärkeiden työsähköpostien seasta. Palautteen antaminen ja sitä kautta keinotekoisien neuroverkon opettaminen voidaan suorittaa esimerkiksi backpropagation-algoritmin avulla. Varsinainen oppiminen tapahtuu keinotekoisessa neuroverkossa olevien neuronien yhteyksien välillä, joiden numeerisia parametreja pyritään optimoimaan backpropagation-algoritmin myötä syötettävän palautteen muodossa.

3.2 Keinotekoisten neuroverkkojen rakenne

Keinotekoisten neuroverkkojen toiminta mielletään usein haastavaksi, koska mustan laatikon -mallin myötä on vaikeampi ymmärtää, mitä tapahtuu, miksi tapahtuu ja miten tapahtuu. Keinotekoisten neuroverkkojen rakenne on pohjimmiltaan kuitenkin lyhyellä perehtymisellä ymmärrettävissä. Professori Kevin Gurney määrittelee lyhykäisyydessään keinotekoisten neuroverkkojen rakenteen seuraavasti: ”Keinotekoinen neuroverkko on yhteen liitetty, yksinkertaisista prosessointielementeistä eli neuroneista koostuva verkosto, jonka toiminta perustuu osittain biologisen hermosolun toiminnan matkimiseen. Neuroverkon prosessointikyky perustuu neuronien välillä olevien yhteyksien vahvuuksiin, jotka voidaan selvittää oppimalla datamallien avulla”. (Gurney, 2004, s. 13)

Ymmärrettäessä se, että keinotekoinen neuroverkko koostuu usein syötekerroksesta, yhdestä tai useammasta piilokerroksesta ja ulostulokerroksesta on neuroverkon rakennetta ja toimintaa huomattavasti helpompi ymmärtää. Rakenteen ja toiminnan kannalta on olennaista tuntea, miten painokertoimet, bias sekä aktivaatiofunktio vaikuttavat yksittäisen neuronin toimintaan ja sitä kautta koko keinotekoisen neuroverkon rakentumiseen ja käyttäytymiseen.



Kuva 3. Keinotekoinen neuroverkko, jossa esiteltyinä syötekerros, piilokerros sekä ulostulokerros. (mukaillen Tuominen, 2017)

3.3 Neuronit

Neuronit ovat keinotekoisessa neuroverkossa peruselementti, jonka rooli keinotekoisien neuroverkkojen rakentumisessa on keskeinen sen ottaessa vastaan tulosignaaleja ja lähettäessä lähtösignaaleja. Neuronit voidaan ajatella professori Gurney'n mukaan eräänlaisena yksinkertaisena prosessointielementtinä, jonka toimintaan painokertoimet, bias sekä aktivaatiofunktio vaikuttavat olennaisesti. Neuronissa lasketaan yhteen painokertoimien ja tulosignaalien tulo, johon lisätään myös vakioparametri. Summattu arvo syötetään aktivaatiofunktioon, jolloin saadaan arvo neuronin lähdölle. Lähtösignaali voi toimia tulosignaalina jopa tuhansille muille neuroneille, jonka myötä keinotekoisien neuroverkkojen kyky prosessoida erilaisia aineistoja kasvaa merkittävästi.

3.3.1 Painokertoimet

Synapseilla tarkoitetaan niitä yhteyksiä, jotka vaikuttavat biologisessa neuroverkossa olevien hermosolujen välillä. Synapsien tarkoituksena on välittää yksittäiseltä neuronilta saatua informaatiota muille neuroverkossa oleville hermosoluille. Painokertoimet ovat keinotekoisessa neuroverkossa vastineita biologisen neuroverkossa synapseille. Kullakin painokertoimella on ominainen matemaattinen lukuarvo, joka kuvaa sitä, kuinka merkittävässä asemassa kyseinen yhteys on eri kerroksissa vaikuttavien neuronien välillä. (Gurney, 2004, s. 13–14) Painokertoimen matemaattisesta suuruudesta riippuen se voi joko lisätä tai vähentää neuronien välillä etenevän signaalin eli aktivaation arvoa. Painokertoimien vaikutus voidaan esittää matemaattisesti muodossa, jossa kerrotaan jokainen yksittäinen tulosignaali sen neuronien välisen yhteyden painokertoimella, jota pitkin signaali etenee. (Gurney, 2004, s. 29)

3.3.2 Bias

Biaseja, b_n , voidaan kuvata eräänlaisiksi vakioparametreiksi, joita lisätään tulosignaalien ja niiden käyttämien painokertoimien summaan, ennen kuin saatu summa syötetään aktivoitavaksi aktivaatiofunktioon. Yleisesti jokaisella piilokerroksessa olevalla neuronilla on oma vakioparametrina toimiva bias. Lisäämällä vakioterminä toimiva bias painotettujen tulosignaalien summaan voidaan vaikuttaa siihen, milloin neuronit aktivoituvat. Biasin voidaan siis katsoa toimivan eräänlaisena vakioarvona, joka täytyy ylittyä, jotta neuronit voi välittää informaatiota seuraavan kerroksen neuroneille. Biasin

merkitys vakioterminä toimimisen lisäksi ilmenee myös silloin, kun on tarve tutkia erilaisten ehdollisten suhteiden merkitsevyyttä. (Thomas, s. 7–8)

3.3.3 Aktivaatiofunktio

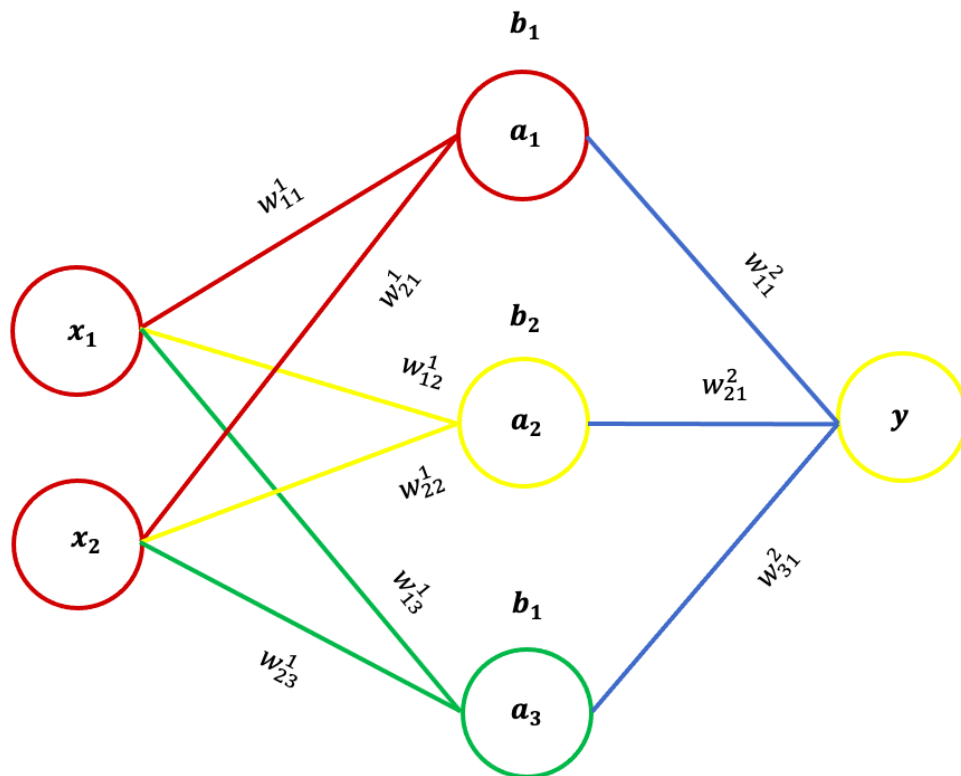
Aktivaatiofunktio on funktio, jonka tarkoituksena on laskea neuronin lähtöpiilokerroksissa tai ulostulokerrokselta lähtevän lähtösignaalin arvo. Aktivaatiofunktion laskema arvo neuronin lähdölle saadaan, kun tarkastellaan painokertoimilla kerrottujen tulosignaalien summaa, johon on huomioitu myös vakio-termi biasin osuus. Aktivaatiofunktion tarkoituksena on luoda epälineaarisuutta keinotekoisien neuroverkon sekaan. Keinotekoisien neuroverkkojen parissa aktivaatiofunktiolta odotetaan usein myös jatkuvaa derivoituvuutta, sillä keinotekoisia neuroverkkoja opettaessa käytetään usein algoritmeja, jotka edellyttävät aktivaatiofunktion derivaatan käyttämistä. Luokitteluongelmien parissa aktivaatiofunktiona käytetään usein logistisena funktiona toimivaa Sigmoid-funktiota, mutta yleisesti luokitteluongelmien ulkopuolella käytettävä aktivaatiofunktio on Rectified Linear Unit eli ReLU-funktio (Starmer, 2020). Yleisesti käytettäessä Sigmoid-funktiota operoidaan suljetulla välillä $\sigma: \mathbb{R} \rightarrow]0,1[$, kun taas ReLU-funktiota käytettäessä $f: \mathbb{R} \rightarrow [0, \infty[$. Aktivaatiofunktiota valittaessa on syytä huomata, että aktivaatiofunktio vaikuttaa keinotekoisien neuroverkon oppimisnopeuteen ja oppimisen vakauteen riippuen siitä, onko käytettävän aktivaatiofunktion väli rajoitettu vai ei. Käytettävän aktivaatiofunktion valintaan vaikuttaa olennaisesti se, millaista funktiota keinotekoisella neuroverkolla on tarkoitus approksimoida (Tuominen, 2017).

Painokertoimet, bias sekä aktivaatiofunktio ovat tekijöitä, jotka mahdollistavat yksittäisen neuronin ja sitä kautta koko keinotekoisien neuroverkon toiminnan. Ymmärrettäessä yksittäisen neuronin rakenne ja sen tekijöiden roolit, on mahdollista ymmärtää suuren keinotekoisien neuroverkon toiminta. Keinotekoinen neuroverkko voi olla rakentunut vaikka miljoonasta neuronista, mutta sen toiminta perustuu suuresta koosta huolimatta siihen, että neuroni vastaanottaa tulosignaalien ja niiden käyttämien painokertoimien tulon, lisää siihen vakio-parametrin arvon ja syöttää summatun arvon aktivoitavaksi aktivaatiofunktion, joka laskee neuronista lähtevän lähtösignaalin arvon. Kuvassa 4 on esitetty, miten keinotekoisessa neuroverkossa vaikuttavat tekijät voivat sijoittua yksinkertaisessa neuroverkossa. Neuronin lähtösignaalin arvo voidaan esittää matemaattisesti kaavalla, joka eräällä tavalla kiteyttää kaikkien neuronin rakenteeseen vaikuttavien osasten merkityksen matemaattiseen muotoon: (Tuominen, 2017)

$$\mathbf{a}_j^l = \varphi(\mathbf{z}_j^l) = \varphi\left(\sum_{i=1}^{N_{l-1}} w_{ij}^l \mathbf{a}_i^{l-1} + b_j^l\right). \quad (1)$$

Yllä,

- a_j^l on kerroksen l neuronin j lähtösignaali eli aktivaation suuruus seuraavalle neuronikerrokselle,
- φ on aktivaatiofunktio,
- z_j^l on kerroksen l neuronin j summa,
- N_l on kerroksen l-1 lukumäärä,
- w_{ij}^l on kerroksen l-1 neuronin i ja kerroksen l neuronin j välinen painokerroin ja
- b_j^l on kerroksen l neuronin j vakiotermi.



Kuva 4. Keinotekoinen neuroverkko, jossa esitettyinä lähdöt, painokertoimet, vakio-termit, aktivaatiofunktioit sekä neuronin ulostulo. (mukaillen, Tuominen, 2017)

3.4 Sigmoid-funktio

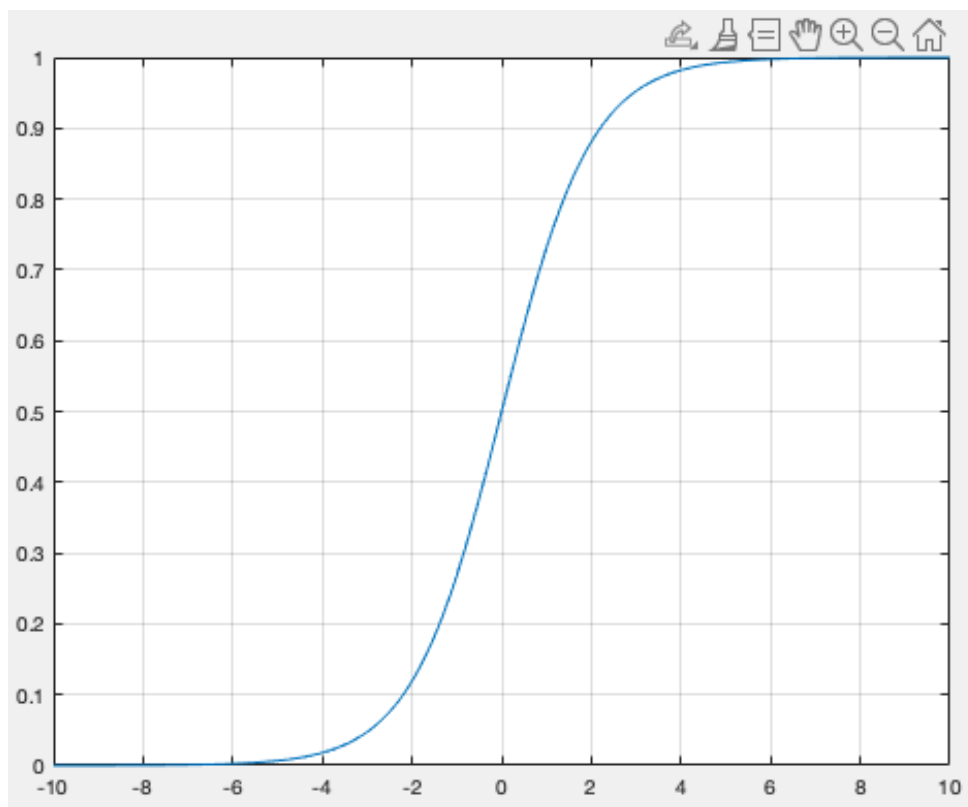
Sigmoid-funktio on rajoitetulla välillä $[0,1]$ arvoja saava jatkuvasti derivoituva, aidosti kasvava logistinen aktivaatiofunktio, joka on yleisesti käytetty aktivaatiofunktio luokitteluongelmissa sekä yleisemmin keinotekoisien neuroverkkojen tuottamien arvojen aktivoimisessa. Suosio esimerkiksi binääristen luokitteluongelmien parissa perustuu siihen, että Sigmoid-funktion saadessa arvoja ainoastaan välillä nolasta yhteen sitä voidaan pitää eräänlaisena kytkimenä, joka kykenee muuttamaan lähtöään sen mukaan, millainen arvo aktivaatiofunktion sisään tulee. Sigmoid-funktion heikkouksina voidaan pitää sitä, että sen reagoituvuus x-akselin saamien arvojen muutoksiin tapahtuu muita aktivaatiofunktioita hitaammin sekä sitä, että Sigmoid-funktion saaman derivaatan arvo on hyvin lähellä arvoa nolla, jos x-akselin saamat arvot ovat hyvin suuria tai vastaavasti hyvin pieniä. Lähellä nollaa oleva derivaatta voi aiheuttaa hankaluuksia myöhemmin muun muassa hitaan oppimisnopeuden muodossa opettaessa keinotekoisia neuroverkkoa luokittelemaan esimerkiksi erilaisia binäärisysteemejä hyödyntäen backpropagation-algoritmiin ja virhefunktion minimoimiseen perustuvia opetusmenetelmiä. Sigmoid-funktiolle on tyypillistä S-muotoinen kuvaaja sekä se, että se sisältää sopivan sekoituksen lineaarista ja epälineaarista käyttäytymistä (Haykin, 2009, s. 14). Sigmoid-funktion matemaattinen kaava on muotoa: (Gurney, 2004, s. 35)

$$y = \sigma(a) \equiv \frac{1}{1 + e^{-\frac{a-\theta}{\rho}}} \quad (2)$$

Yllä,

- y on Sigmoid-funktiolla aktivoitu neuroni,
- e on Neperin luku, luonnollisten logaritmien kantaluku, arvoltaan noin 2,7183,
- ρ määrää Sigmoid-funktion muodon (suuret arvot tekevät käyrästä tasaisemman ja pienet jyrkemmän) ja
- θ kuvaa Sigmoid-funktion paikkaa x-akselilla.

Usein parametrille ρ käytetään arvoa 1.



Kuva 5. Sigmoid-funktion tuottama kuvaaja ($\rho = 1$ ja $\theta = 0$). Kuvaaja on tyypillinen S-muotoinen kuvaaja, joka reagoi verkkaisesti x-akselin saamien arvojen muutoksiin.

3.5 Keinotekoisten neuroverkkojen opettaminen

Keinotekoisten neuroverkkojen opettamisen tavoitteena on löytää neuroverkossa oleville parametreille optimaaliset arvot. Opettamisen onnistuminen määräytyy sen mukaan, kuinka hyvin neuroverkon parametrit eli neuronien väliset painokertoimet sekä vakioarvot eli biasit saadaan optimoituksi esimerkiksi tietynlaisen luokitteluongelman ratkaisemiseksi. Parametrien optimointi tapahtuu iteroinnin kautta eli menetelmällä, jossa tiettyjä työvaiheita toistetaan niin kauan, kunnes haluttu lopputulos saavutetaan tai vastaavasti työvaiheiden toistaminen päätetään lopettaa. Jokaisen iteroinnin jälkeen neuroverkon painokertoimiin ja vakioarvoihin tehdään pieniä muutoksia sen mukaan, mihin suuntaan neuroverkon luokittelukyky on edennyt halutusta suunnasta. Menetelmät, joissa neuroverkon tuottamia tuloksia valvotaan ja sen kanssa vaikutetaan vuorovaikuttaisesti, kutsutaan ohjatuksi oppimiseksi. (Gurney, 2004, s. 63)

Ohjatun oppimisen yksi lähestymistapa perustuu neuroverkon todellisen lähdön ja kohdevektorin välisen eron määrittelyyn, jossa eroa pyritään minimoimaan ikään kuin virheenä säätämällä neuroverkossa olevien parametrien arvoja. Käsiteltäessä lähdön

ja kohdevektorin välistä eroa virheenä yritetään löytää virheiden vähimmäismäärä, jossa virheiden katsotaan olevan riippuvaisia neuroverkon parametrien funktiona. (Gurney, 2004, s. 81) Virheiden vähimmäismäärää voidaan yrittää etsiä gradienttimenetelmän avulla.

3.5.1 Gradienttimenetelmä

Gradientilla tarkoitetaan yleisesti differentiaalilaskennassa käytettävää matemaattista suuretta, joka ilmaisee tietyn funktion suurimman muutosnopeuden sekä kertoo sen, että mihin suuntaan muutoksen aiheuttamat voimat ajavat funktiota. Gradienttimenetelmässä pyritään selvittämään syötearvon x ja tavoitearvon y välinen virhefunktion gradientti. Virhefunktion gradientti toimii normaalin gradientin tapaan eli sen avulla voidaan selittää muutosnopeudet sekä suunnat, kun neuroverkon parametrien arvot muuttuvat. Minimoitaessa syötearvon x ja tavoitearvon y välistä virhefunktiota pyritään olennaisesti löytämään virhefunktion gradientille sellaiset arvot, joilla virhefunktion gradientin arvo vähenisi voimakkaasti. Gradienttimenetelmää käytettäessä pyritään siis siihen, että virhefunktion suuruus saadaan minimoitua. Virhefunktion minimoiminen perustuu siihen, että neuroverkkoa opetetaan syötteisiin ja tavoitteisiin liittyvien arvojen avulla. Virhefunktion minimoiminen muistuttaa jossain määrin pienimmän neliösumman menetelmää, jota hyödynnetään matemaattisesti virhefunktion kaavassa:

$$E = \frac{1}{2} |t - y|^2 = \frac{1}{2} \sum_{k=1}^m (t_k - y_k)^2. \quad (3)$$

Yllä,

- E on virhefunktion arvo,
- t_k on neuroverkon syötteelle k antama tulos ja
- y_k on neuroverkon tuottaman tuloksen tavoitearvo.

Tällöin neuroverkossa olevia parametreja pyritään siis optimoimaan niin, että neuroverkon syötteelle antaman tuloksen x ja neuroverkon tavoitearvon y välillä on mahdollisimman pieni erotus. Gradienttimenetelmää käytettäessä virhefunktion E arvo

lasketaan tietylle esimerkkijoukolle eli tietyin parametrin suhteen. Gradientilla viitataan tässä yhteydessä virhefunktion E osittaisderivaattoihin neuroverkossa esiintyvien painokertoimien ja biasien eli vakioarvojen suhteen eli osittaisderivaattojen $\frac{\partial E}{\partial w}$ ja $\frac{\partial E}{\partial b}$ suhteen. Kaikkien neuroverkossa olevien parametrien vaikutus virheeseen saadaan selville yksittäisten virheiden sijaan käyttämällä apuna opettamisessa backpropagation-algoritmia. (Tuominen, 2017)

3.5.2 Backpropagation-algoritmi

Backpropagation-algoritmin toiminta perustuu siihen, että keinotekoisien neuroverkon mallinnusvirhe syötetään syötekerroksen sijaan ulostulokerroksesta lähtien, jolloin signaalit etenevät vastavirtaan neuroverkossa. Backpropagation-algoritmin käytön tavoitteena on löytää neuroverkossa oleville parametreille optimaalisimmat arvot minimoimalla neuroverkon tuottamien lähtöjen virhefunktion suuruutta. Virhefunktiota minimoimalla parametrit saavat optimaalisempia arvoja ja tällöin keinotekoisien neuroverkon luokittelukyky tehostuu. Perusidea on siis sama, kuin gradienttimenetelmässä, mutta backpropagation-algoritmia käytettäessä optimoidaan useampia kerroksia yhtä aikaa. (Gurney, 2004, s. 97–98).

Backpropagation-algoritmia käytettäessä ensimmäinen tehtävä on syöttää parametreille jotkin satunnaiset arvot, joilla neuroverkko tuottaa ensimmäisen tuloksensa. Annettaessa ensimmäisiä satunnaisia arvoja neuroverkolle on huomioitava, että keinotekoinen neuroverkko tuottaa yleisesti arvoja väliltä nolosta yhteen, joten minkään neuronin lähdön ei tulisi saada lukuarvoa läheltä 0 tai 1, koska neuroni on tällöin vaarassa satureitua. Alkuarvojen ollessa annettuina ja ensimmäisen tuloksen ollessa selvillä backpropagation-algoritmin avulla voidaan lähteä optimoimaan parametrien arvoja iteroinnin avulla. Iteroinnilla voidaan viitata siihen tekemiseen, jossa tiettyjä työvaiheita toistetaan niin kauan, kunnes haluttu lopputulos saavutetaan tai vastaavasti iterointi päätetään lopettaa esimerkiksi tuloksettomana. (Gurney, 2004, s. 101)

Backpropagation-algoritmia voidaan ajatella eräänlaisena silmukkarakenteena, jonka hyödyntäminen noudattaa pääpiirteittäin samoja välivaiheita. Optimointia suoritetaan iteroinnin kautta niin kauan, että haluttu tulos saavutetaan tai iteroiminen päätetään lopettaa. Tyypillinen luokitteluongelma on esimerkiksi kuvantunnistukseen liittyvät

ongelmat, joiden yhteydessä menetellään kutakuinkin seuraavasti: (Gurney, 2004, s. 102–103)

1. Esitetään data-aineisto syötekerrokselle.
2. Odotetaan, että piilokerrokset käsittelevät saamaansa aineistoa, jonka perusteella tekevät ensimmäiset optimointinsa painokertoimille ja vakioarvoille.
3. Odotetaan, että ulostulokerroksen neuronit saavat arvioitua piilokerrosten tuottamaa informaatiota.
4. Syötetään tavoiteaineisto neuroverkon käsiteltäväksi ulostulokerroksen kautta, jolloin neuroverkko toimii vastavirtaan.
5. Lasketaan ulostulokerroksen tuottaman syötteen tuloksen ja neuroverkon tavoitearvon välinen virhefunktio.
6. Optimoidaan ulostulokerroksen parametreja gradienttimenetelmän avulla.
7. Optimoidaan piilokerroksien parametreja gradienttimenetelmän avulla.

3.6 Keinotekoisia neuroverkkoja hyödyntäviä luokittelutilanteita

Keinotekoisia neuroverkkoja ja niiden kykyä luokitella erilaisia aineistoja käytetään laajalti eri sovellusten yhteydessä. Neuroverkkojen käyttö ilmenee etenkin termin syväoppiminen kautta, jolla viitataan yleisesti useamman kerroksen keinotekoiseen neuroverkkoon. Tyypillisiä luokitteluongelmia, joihin keinotekoisien neuroverkkojen kautta pyritään saamaan apua ovat erilaiset kuvan-, tekstin- ja puheentunnistukseen liittyvät ongelmat. Eräs arkipäivän esimerkki luokittelutilanteesta on esimerkiksi älypuhelimien suojalukituksen avaaminen kasvojen avulla. Älypuhelin voidaan opettaa tunnistamaan puhelimen omistaja esimerkiksi iirisskannauksen avulla, jolloin älypuhelin oppii luokittelemaan, millaisille silmän väriä väreille suojalukitus avataan. Keinotekoisien neuroverkkojen luokittelukykyä hyödynnetään terveydenhuollossa esimerkiksi luokitella pahalaatuisia luomia hyvälaatuisien seasta, kun taas puheentunnistusta hyödynnetään esimerkiksi erilaisten tulkkauspalveluiden yhteydessä. Tekstintunnistusta hyödynnetään esimerkiksi erilaisten hakukonepalveluiden tarjoamien ehdotuksien parissa. Hyvä esimerkki tekstiin liittyvästä tunnistuksesta on hakukonepalvelu Googlen tarjoamat hakutulokset vain muutaman kirjaimen perusteella. Professori Gurney esittelee kaksi mainiota luokitteluun perustuvaa esimerkkiä teoksessaan, jotka ovat seuraavat:

1. Keinotekkoisten neuroverkkojen luokittelukykyä hyödynnetään lääketieteellisten diagnoosien yhteydessä erilaisten hoitajaksojen pituutta arvioitaessa. Esimerkiksi psykiatrista hoitoa vaativan potilaan hoitajakson pituutta arvioitaessa keinotekoinen neuroverkko luo arvion hoitajakson pituudesta luokittelun avulla saadessaan tietoa potilaan demografisista, sosiaalisista ja kliinisistä tiedoista. Tietojen perusteella neuroverkko luo suuntaa antavan arvion siitä, kuinka kauan potilaan olisi hyvä olla hoidon piirissä sekä arvion siitä, millaiset potilaan henkilökohtaiset resurssit tulevat olemaan hoidon päätyttyä. (Gurney, 2004, s. 126)
2. Keinotekkoisten neuroverkkojen luokittelukykyä hyödynnetään ennustettaessa eri pörssiyhtiöiden tarjoamien osakkeiden mahdollisia tuottoja pääomamarkkinoilla. Neuroverkkoa opetetaan luokittelemaan kolmen parametrin, tässä tapauksessa A, B ja C avulla yrityksen tarjoaman osakkeen suorituskykyä kuuden kuukauden päähän, tutkimalla yrityksen tarjoamia taloudellisia raportteja eli taseita. Neuroverkko oppii tunnistamaan ja luokittelemaan menestyksekkäiden parametrien arvot, joiden kautta se luo positiiviset tulevaisuudennäkymät tiettyjen yritysten osakkeille. (Gurney, 2004, s. 127-128)

Vaikka neuroverkko olisi opetettu laadukkaalla aineistolla ammattilaisten toimesta ja neuroverkossa olevat parametrit olisi saatu niin optimaalisiksi kuin mahdollista niin on muistettava, että keinotekoinen neuroverkko on lopulta kuitenkin vain yksinkertaisista prosessointisoluihin koostuva verkosto, johon ei tule sokeasti luottaa. Esimerkiksi osakemarkkinoilla osakkeiden hintaan vaikuttaa hyvin moni tekijä, joten neuroverkon tuottama luokittelu niin sanottuihin kannattaviin ja ei-kannattaviin osakkeisiin ei ole täysin validi. Osakemarkkinoilla osakkeen hintaa vaikuttavat muun muassa hallitusten päätökset, talouspolitiikka sekä yleinen luottamus markkinoiden toimivuuteen (Gurney, 2004, s. 129). Neuroverkkojen tuottamia tuloksia ei siis tule pitää esimerkiksi minään sijoitusneuvoina.

4 NEUROVERKKOLUOKITTIMEN MUODOSTAMINEN

4.1 Kokeellisen osion tavoite ja valitun aineiston esittely

Tämän kandidaatintyön kokeellisen osion tarkoituksena on esitellä yleisluonteisesti yhtä mahdollista tapaa, jolla neuroverkkoluokitin saadaan muodostettua tietyn luokitteluongelman ratkaisemiseksi hyödyntäen numeerisen laskennan tietokoneohjelmisto MATLAB®:ia. Työn tarkoituksena ei ole opettaa MATLAB®:lla ohjelmointia, mutta käytettyjä komentoja selitetään auki hyvin lyhyesti kokeellisen osion edetessä. Kokeellisessa osiossa keinotekoisia neuroverkkoja tullaan opettamaan laajalti data-analytiikan ja keinotekoisien neuroverkkojen parissa opettamiseen käytetyn tietokannan avulla. Valitun aineiston pohjalta pyritään muodostamaan keinotekoisia neuroverkkoja hyödyntäen sellainen neuroverkkoluokitin, joka havainnollistaa sitä luokittelukykyä, joka keinotekoisille neuroverkoille voidaan saada. Yleisesti ottaen neuroverkkoluokittimen muodostaminen etenee seuraavasti:

- aineiston kerääminen,
- aineiston esikäsittely,
- mallin muodostaminen ja
- mallin hyvyyden arviointi.

Tähän työhön on valittu valmiiksi aineistoksi Kaggle-datapankista Mnist-tietokanta. Kyseinen aineisto sisältää yhteensä 70 000 käsin kirjoitettua kokonaislukua, jotka ovat väliltä 0–9. Aineisto on jaettu siten, että aineistosta voidaan käyttää 60 000 kohtaa keinotekoisien neuroverkon opettamiseen ja 10 000 kohtaa keinotekoisien neuroverkon luokittelukykyyn testaamiseen. Opetettaessa keinotekoisia neuroverkkoja Mnist-tietokannan avulla opettamisen tarkoituksena on se, että keinotekoiselle neuroverkolle saataisiin sellainen luokittelukyky, että se pystyisi luokittelemaan aineistossa olevat muuttujat niiden todellisiin luokkakategorioihin.

4.2 Aineiston esikäsittely

Aineiston kerääminen ei tässä työssä ole merkityksellistä, koska työhön on valittu valmis aineisto Mnist-tietokannan muodossa. Aineistoa on kuitenkin hyvä esikäsitellä, jolloin voidaan varmistaa se, että aineisto on käyttökelpoista neuroverkkoluokittimen

muodostamiseen. Ladattaessa Mnist-tietokantaa Kaggle-datapankista koostuu se kahdesta CSV-tiedostosta. CSV-tiedostoille on ominaista se, että taulukkomuotoista informaatiota tallennetaan tekstitiedoston muotoon.

Esikäsittelyn ensimmäinen vaihe on poimia talteen Mnist-tietokannan keinotekoisten neuroverkkojen opettamiseen käytettävästä opetusaineistosta tulo- ja lähtömuuttujat, jotka saadaan seuraavilla komentoikkunaan kirjoitettavilla komennoilla:

```
>> output = mnisttrain(:,1);
>> input = mnisttrain(:,2:end);
```

Yllä,

- output on lähtömuuttuja opetusaineistossa,
- mnisttrain on käsiteltävä neuroverkon opetuskäyttöön tarkoitettu tietokanta,
- (:,n) on komento, jolla valitaan kaikki rivin n alkiot,
- input on tulomuuttuja opetusaineistossa ja
- kaksoispilkun avulla estetään koodin välivaiheita näkymästä. Ei vaikuta millään tavoin saataviin tuloksiin eli voidaan jättää huomiotta.

Seuraava tehtävä on pienentää aineistoa siltä osin, että sellaiset muuttujat, joiden keskihajonta on nolla, poistetaan työstä. Tilanteessa, jossa keskihajonta on nolla, muuttuja saa tilanteesta riippumatta aina saman arvon, jolloin se ei sisällä mielekäästä informaatiota. Muuttujat, joiden keskihajonta on nolla, saadaan poistettua seuraavilla komennoilla:

```
>> ind1 = std(input) == 0;
>> input(:,ind1) = [];
```

Yllä,

- ind1 on lisämääritelty muuttuja, jolle määritellään sellaiset muuttujat, joiden keskihajonta on nolla,

- std on komento, joka laskee kaikkien tulomuuttujien seasta ne arvot, joiden keskihajonta on nolla ja
- [] on eräänlainen tyhjä matriisi, johon keskihajonnan nolla omaavat arvot määritellään kuuluvaksi. Voidaan jättää huomiotta, koska ne eivät tarjoa mielekästä informaatiota.

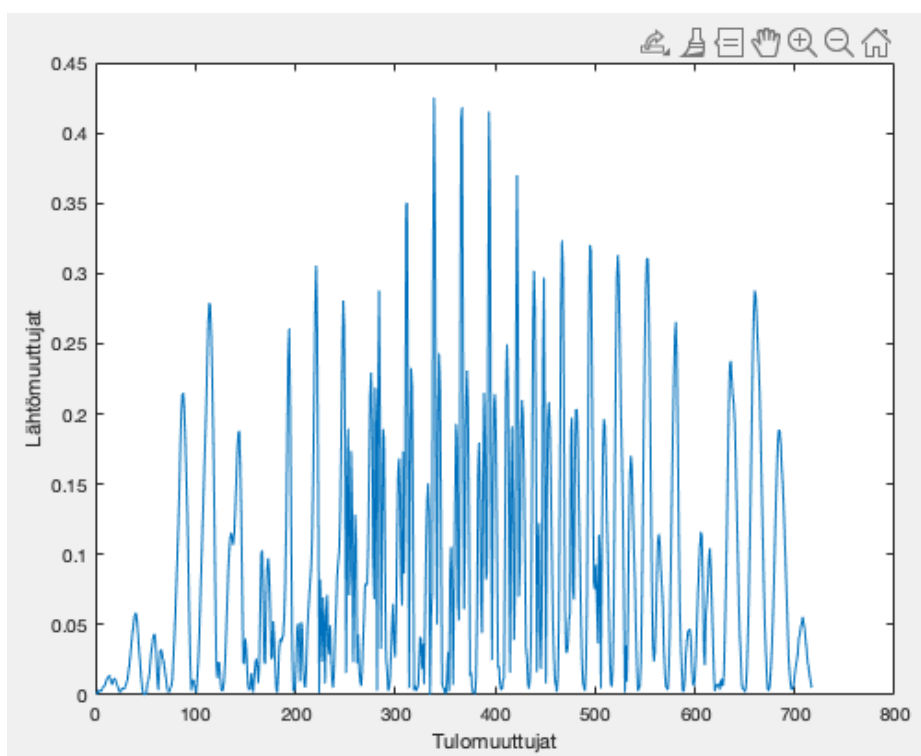
Opetusaineistosta on nyt poimittu talteen tulo- ja lähtömuuttujat sekä aineistosta on poistettu sellaiset muuttujat, jotka eivät tarjoa mielekästä informaatiota. Havainnollistetaan nyt käsiteltävää aineistoa tutkien sitä, millainen korrelaatio aineiston tulo- ja lähtömuuttujien välillä vaikuttaa. Korrelaatio saadaan selville seuraavilla komennoilla:

```
>> r = corr(output, input);  
>> plot(abs(r))  
>> xlabel('Tulomuuttujat')  
>> ylabel('Lähtömuuttujat')
```

Yllä,

- r on muuttuja,
- corr on komento, jolla saadaan tutkittua lähtö- ja tulomuuttujan välistä korrelaatiota,
- plot on komento, joka mahdollistaa kuvaajan piirtämisen,
- abs on komento, joka palauttaa muuttujan r absoluuttisten elementtien arvot,
- xlabel on komento, jolla saadaan vaaka-akselin otsikko ja
- ylabel on komento, jolla saadaan pystyakselin otsikko.

Tulo- ja lähtömuuttujien välinen korrelaatio nähdään kuvasta 6:



Kuva 6. Tulo- ja lähtömuuttujien välistä korrelaatiota havainnollistava kuvaaja.

Korrelaation ollessa havainnollistettuna poistetaan aineistosta seuraavaksi muuttujat, joiden korrelaation itseisarvo on < 0.2 . Oletuksena tällaiset muuttujat halutaan poistaa, koska tällöin aineiston kokoa saadaan pienennettyä ilman, että menetetään työn kannalta mahdollisesti olennaista tietoa. Halutut muuttujat saadaan poistettua nyt komennolla:

```
>> ind2 = abs(r) < 0.2;
>> input(:,ind2) = [];
```

Yllä,

- ind2 on lisämääritelty muuttuja, jolla poistetaan aineistosta muuttujat, joiden korrelaation itseisarvo on < 0.2 ja
- [] on eräänlainen tyhjä matriisi, johon määritellään ne muuttujat, joiden korrelaation itseisarvo on < 0.2 . Voidaan jättää huomiotta, koska ne eivät tarjoa mielekästä informaatiota.

Suoritettujen esikäsittelyjen jälkeen ollaan tilanteessa, jossa opetusaineiston tulo- ja lähtömuuttujat on poimittu talteen sekä aineistosta on poistettu sellaiset muuttujat, joiden keskihajonta on nolla ja muuttujat, joiden korrelaation itseisarvo on pienempi, kuin 0.2.

Mnist-tietokanta sisältää erilliset keinotekoisille neuroverkoille tarkoitetut opetus- ja testausaineistot, joten suoritetaan seuraavaksi testausaineistolle samat toimenpiteet:

```
>> outputtest = mnisttest(:,1);
>> inputtest = mnisttest(:,2:end);
>> inputtest(:,ind1) = [];
>> inputtest(:,ind2) = [];
```

Yllä,

- outputtest on lähtömuuttuja testausaineistossa,
- mnisttest on neuroverkon testaamiseen tarkoitettu tietokanta,
- inputtest on tulomuuttuja testausaineistossa

Viimeinen vaihe, joka Mnist-aineistolle täytyy suorittaa, on muuttaa lähtömuuttujat dummy-muuttujiksi. Lähtömuuttujien muuttaminen onnistuu seuraavilla komennoilla:

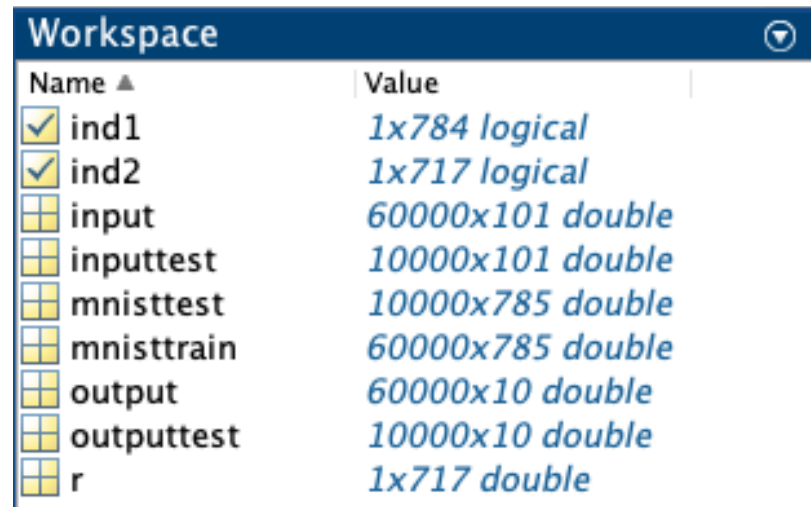
```
>> output = categorical(output);
>> output = dummyvar(output);
>> outputtest = categorical(outputtest);
>> outputtest = dummyvar(outputtest);
```

Yllä,

- categorical on määrittely, jonka avulla saadaan jaettua lähtömuuttujat luokkiin. Tässä tapauksessa haluttuja ryhmiä ovat Mnist-tietokannan sisältämät kokonaisluvut väliltä 0–9 eli halutaan muodostaa 10 luokkaa ja
- dummyvar on komento, jonka tarkoituksena on muuttaa lähtömuuttuja kymmeniksi dummy-muuttujaksi. Tavoitteena on muokata lähtömuuttujaa niin, että yksi sarake sisältää kaikki Mnist-tietokannassa olevat 10 muuttujaa, jotka saavat ainoastaan joko arvoja 0 tai 1.

Suoritetuilla esikäsittelyillä ollaan nyt vaiheessa, jossa alkuperäisistä kahdesta CSV-muotoisesta tiedostosta on saatu muodostettua tarvittavat neljä datamatriisia, joilla keinotekoinen neuroverkkoluokitin voidaan muodostaa. Tarvittavat datamatriisit ovat opetusvaiheessa käytettävät $60\,000 \times 101$ kokoinen input -datamatriisi ja $60\,000 \times 10$

kokoinen output -datamatriisi sekä testausvaiheessa käytettävät $10\,000 \times 101$ kokoinen inputtest -datamatriisi ja $10\,000 \times 10$ kokoinen outputtest -datamatriisi.

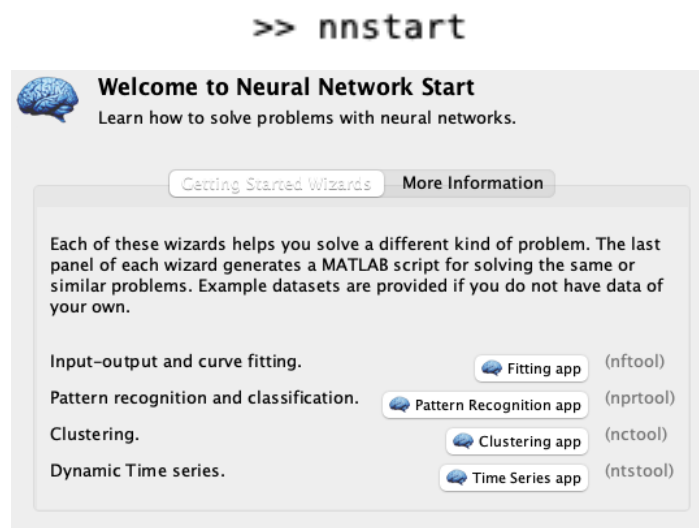


Name ▲	Value
<input checked="" type="checkbox"/> ind1	1x784 logical
<input checked="" type="checkbox"/> ind2	1x717 logical
<input type="checkbox"/> input	60000x101 double
<input type="checkbox"/> inputtest	10000x101 double
<input type="checkbox"/> mnisttest	10000x785 double
<input type="checkbox"/> mnisttrain	60000x785 double
<input type="checkbox"/> output	60000x10 double
<input type="checkbox"/> outputtest	10000x10 double
<input type="checkbox"/> r	1x717 double

Kuva 7. Esikäsittelyn aikana työtilaan luodut ja tuodut muuttujat. (MATLAB®, R2020b)

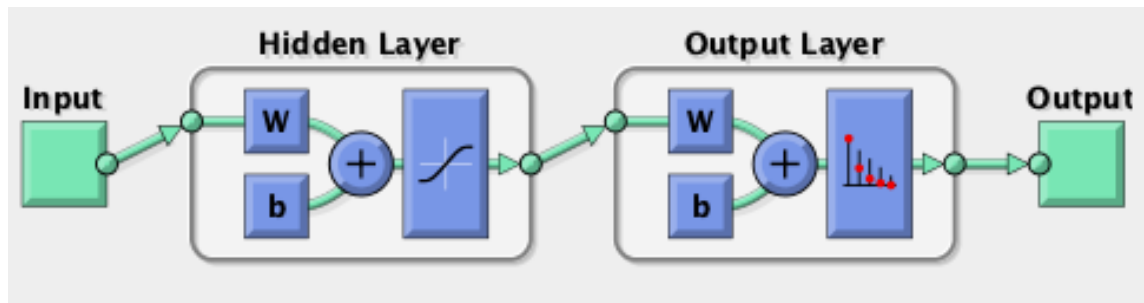
4.3 Mallin muodostaminen

Neuroverkkoluokittimen muodostaminen MATLAB®:ssa edellyttää neural network -toolboxin asentamista. Kyseisen lisäosan ollessa ladattuna neuroverkkoluokittimen muodostaminen aloitetaan kirjoittamalla komentoikkunaan seuraava komento, joka avaa seuraavan ikkunan:



Kuva 8. Neuroverkkoluokittimen valinta (MATLAB®, R2020b)

Tämän työn aihe on keinotekoisten neuroverkkojen käytettävyys luokitteluongelmissa ja kokeelliseen osioon valittu Mnist-tietokanta on eräänlainen kuviontunnistusta vaativa luokitteluongelma, joten neuroverkkoluokittimen muodostaminen aloitetaan valitsemalla Pattern Recognition app. Kuviontunnistusta vaativissa luokitteluongelmissa keinotekoinen neuroverkko halutaan opettaa luokittelemaan aineistoa niiden todellisiin luokkiinsa. Neuroverkkoluokitin muodostetaan hyödyntäen kuvan 9 keinotekoista neuroverkkoa.

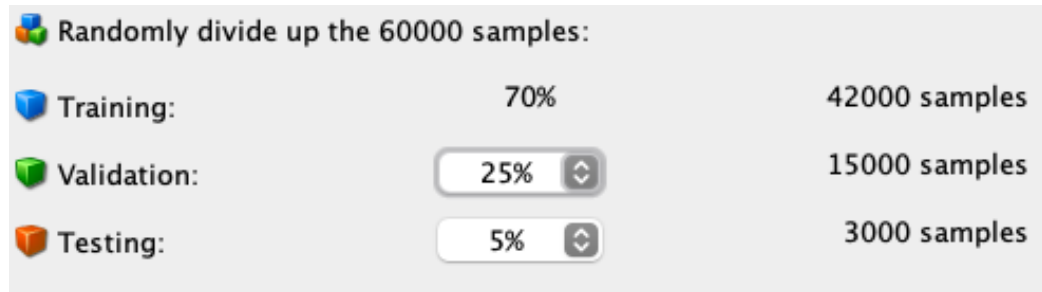


Kuva 9. Neuroverkkoluokittimen muodostamiseen käytettävä keinotekoinen neuroverkko (MATLAB®, R2020b)

Neuroverkkoluokittimen muodostaminen aloitetaan syöttämällä käsiteltävä aineisto keinotekoisten neuroverkkojen tutkittavaksi sekä määrittelemällä syötetylle aineistolle tavoitetilä, johon pyritään. Keinotekoille neuroverkoille syötetään käsiteltäväksi esikäsittelyssä saatu $60\,000 \times 101$ kokoinen input -datamatriisi ja tavoiteaineistoksi myöskin esikäsittelyssä saatu $60\,000 \times 10$ kokoinen output -datamatriisi. Molemmilla matriiseilla on 60 000 riviä, sillä Mnist-tietokannassa aineisto koostui 60 000 käsin kirjoitetusta numerosta. Output -datamatriisissa on 10 saraketta, joista jokainen kuvaa yhtä muuttujaa eli mahdollista luokkaa, joihin keinotekoinen neuroverkko suorittaa luokittelut. Output -datamatriisiin yhteydessä on hyvä huomata se, että muuttujat eli aineistossa olevien kokonaislukujen mahdolliset luokat ovat sarakkeilla.

Seuraavaksi keinotekoille neuroverkoille syötettävä aineisto on jaettava kolmeen osaan. Opetusaineistoa syötetään keinotekoisten neuroverkkojen käsiteltäväksi opetusvaiheessa, jolloin pyritään optimoimaan neuroverkossa vaikuttavia vapaita parametreja. Validointiaineiston avulla seurataan neuroverkon oppimiskyvyn kehittymistä. Testausaineisto ei vaikuta opetusvaiheeseen millään tavoin vaan kertoo lähinnä siitä, miten keinotekoinen neuroverkko on toiminut ennen ja jälkeen suoritetun opetusvaiheen.

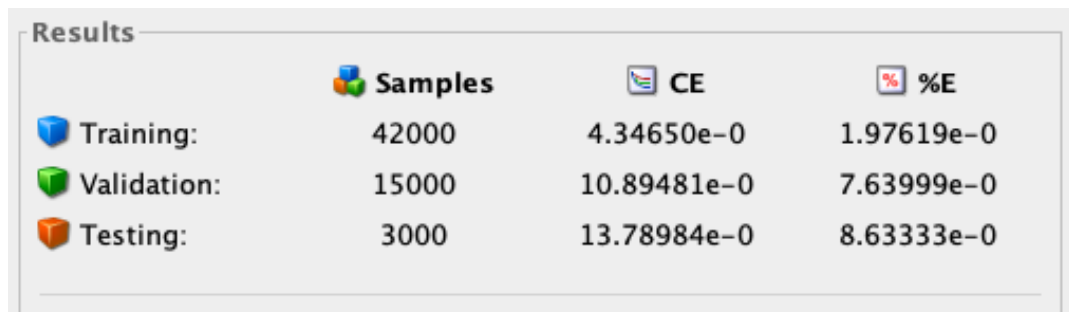
Kuvassa 10 on esitetty suhteet, joilla neuroverkkoluokitinta tässä yhteydessä muodostetaan.



Randomly divide up the 60000 samples:			
Training:	70%		42000 samples
Validation:	25%		15000 samples
Testing:	5%		3000 samples

Kuva 10. Aineiston jako opetus-, validointi- ja testausaineistoon. (MATLAB®, R2020b)

Aineiston jakamisen jälkeen voidaan aloittaa iterointia hyödyntäen selvittämään sitä, mikä on se määrä piilokerroksessa olevia neuroneja, joilla keinotekoinen neuroverkko suorittaa luokittelun halutulla tarkkuudella. Iteroinnilla viitataan siihen tekemiseen, jossa tiettyä työvaihetta suoritetaan niin kauan, kunnes haluttu tarkkuus saavutetaan tai iterointi päätetään lopettaa. Iteroinnilla voidaan siis kokeilla esimerkiksi, tarjoaako 70, 600 vai 2000 piilokerroksen neuronien parhaan tarkkuuden. Kuvassa 11 esitetään esimerkiksi saadut tulokset, kun piilokerroksessa käytettiin 600 neuronien:

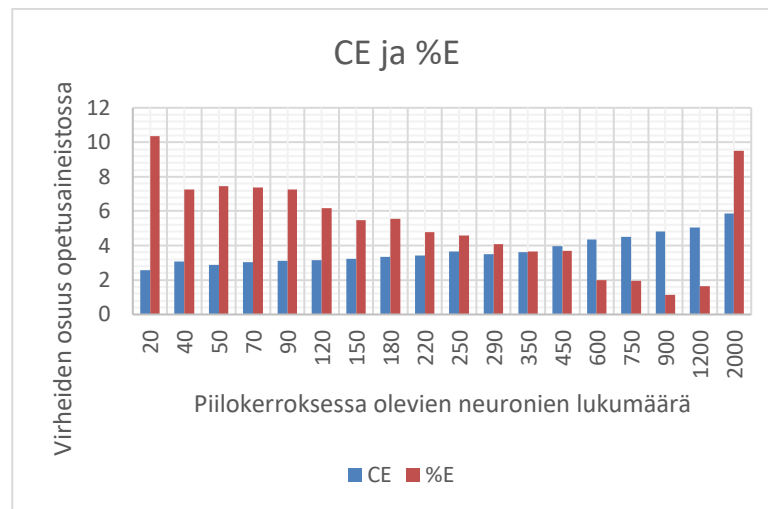


	Samples	CE	%E
Training:	42000	4.34650e-0	1.97619e-0
Validation:	15000	10.89481e-0	7.63999e-0
Testing:	3000	13.78984e-0	8.63333e-0

Kuva 11. 600 piilokerroksen neuronilla saadut tulokset. (MATLAB®, R2020b)

CE viittaa termiin Cross-Entropy, joka on eräänlainen luokittelun hyvyttä arvioitaessa käytettävä mitta. CE-luku ilmaisee jonkin tietyn muuttujan tai tapahtumasarjan kahden todennäköisyysjakauman välistä eroa. Mitä pienempi todennäköisyysjakaumien välinen ero on niin sitä tarkempia tulokset ovat. %E puolestaan kuvaa suhteellista mallinnusvirhettä. Mitä suurempi suhteellisen mallinnusvirheen saama arvo on, niin sitä enemmän keinotekoinen neuroverkko on luokitellut aineistoa väärin luokkakategorioihin. Luokitteluongelmien yhteydessä pyritään minimoimaan sekä CE-luvun, että %E-luvun suuruutta. Kuvassa 12 on esitetty 42 000 näytettä sisältävästä

opetusaineistosta löytyneiden virheiden osuudet. Pylväsdiagrammia katsoessa huomaa, että piilokerroksessa olevien neuronien lukumäärän kasvaessa CE-luku pääsääntöisesti kasvaa, mutta samalla suhteellisen virheen, %E, osuus pienenee:



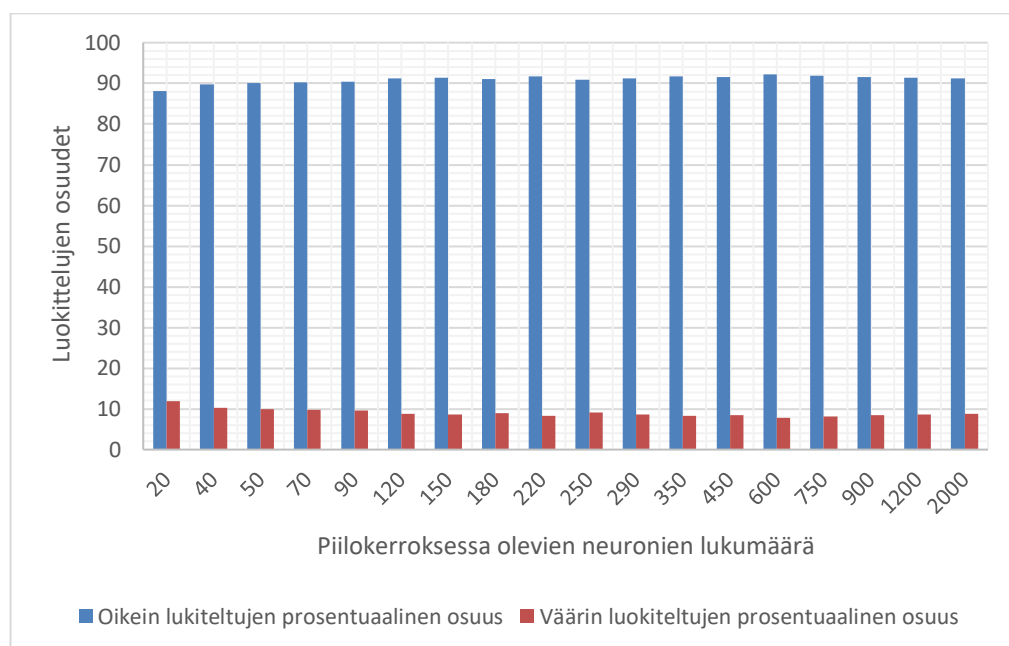
Kuva 12. Opetusaineiston virheiden osuudet erikokoisille piilokerroksille. (MATLAB®, R2020b)

Keinotekoisien neuroverkon luokittelukykyä havainnoitiin opetusaineiston perusteella, jonka jälkeen sen luokittelukykyä voidaan arvioida eri keinoin. Yksi vaihtoehto on vaihdella piilokerroksessa olevien neuronien lukumäärää, jolloin saavutetaan erilaisia luokittelutarkkuuksia. Toinen vaihtoehto on testata neuroverkon toimivuutta testausaineiston avulla. Esikäsittelyn tuloksena saatiin neljä datamatriisia, joista inputtest- ja outputtest-datamatriiseja ei ole vielä hyödynnetty. Mnist-tietokannan sisältämä testausaineisto sisältää 10 000 kuvaa, joiden avulla neuroverkon toimivuutta voidaan selvittää. Syötetyn testausaineiston perusteella MATLAB®:in tarjoama neuroverkkoluokittimen viritysovellus tarjoaa sille syöteyllä datalle kuvassa 13 esitetyn matriisin. Matriisissa rivit vastaavat ennustettuja luokkia Mnist-tietokannan numeroille, kun taas sarakkeet puolestaan vastaavat todellisia luokkia. Vihreällä pohjalla olevat lävistäjän suuntaisesti olevat solut vastaavat tilanteita, joissa neuroverkko on kyennyt luokittamaan numeron sen todelliseen luokkaansa. Matriisin oikean alakulman solu kokoaa koko matriisin tiedot yhteen ja esittää prosentiosuuksina onnistuneet- ja epäonnistuneet luokittelut:

Output Class \ Target Class	1	2	3	4	5	6	7	8	9	10	Accuracy
1	952 9.5%	2 0.0%	10 0.1%	2 0.0%	3 0.0%	7 0.1%	13 0.1%	0 0.0%	4 0.0%	3 0.0%	95.6%
2	1 0.0%	1111 11.1%	6 0.1%	3 0.0%	1 0.0%	4 0.0%	5 0.1%	8 0.1%	0 0.0%	4 0.0%	97.2%
3	6 0.1%	8 0.1%	938 9.4%	15 0.1%	7 0.1%	4 0.0%	6 0.1%	13 0.1%	13 0.1%	4 0.0%	92.5%
4	0 0.0%	3 0.0%	8 0.1%	913 9.1%	2 0.0%	28 0.3%	3 0.0%	8 0.1%	16 0.2%	7 0.1%	92.4%
5	1 0.0%	2 0.0%	7 0.1%	1 0.0%	902 9.0%	1 0.0%	13 0.1%	23 0.2%	6 0.1%	51 0.5%	89.6%
6	10 0.1%	1 0.0%	7 0.1%	39 0.4%	1 0.0%	803 8.0%	5 0.1%	2 0.0%	15 0.1%	6 0.1%	90.3%
7	5 0.1%	3 0.0%	4 0.0%	0 0.0%	8 0.1%	18 0.2%	905 9.0%	0 0.0%	10 0.1%	1 0.0%	94.9%
8	2 0.0%	2 0.0%	15 0.1%	8 0.1%	17 0.2%	4 0.0%	0 0.0%	928 9.3%	9 0.1%	44 0.4%	90.2%
9	1 0.0%	3 0.0%	31 0.3%	24 0.2%	5 0.1%	18 0.2%	5 0.1%	5 0.1%	895 8.9%	13 0.1%	89.5%
10	2 0.0%	0 0.0%	6 0.1%	5 0.1%	36 0.4%	5 0.1%	3 0.0%	41 0.4%	6 0.1%	876 8.8%	89.4%
Overall	97.1%	97.9%	90.9%	90.4%	91.9%	90.0%	94.5%	90.3%	91.9%	86.8%	92.2%
Overall	2.9%	2.1%	9.1%	9.6%	8.1%	10.0%	5.5%	9.7%	8.1%	13.2%	7.8%

Kuva 13. Testausaineistolla saatava neuroverkon luokittelukykyä kuvaava matriisi 600 neuronilla piilokerroksessa. (MATLAB®, R2020b)

Kuvassa 14 on esitetty kahdeksallatoista erilaisella piilokerroksessa olevien neuronien lukumäärällä saadut tulokset sille, kuinka suuret prosentuaaliset osuudet kullakin iteroinnilla saatiin onnistuneille ja epäonnistuneille testausaineiston luokitteluille:



Kuva 14. Neuroverkkoluokittimen testausaineiston perusteella luokittelemat onnistuneet ja epäonnistuneet luokittelut. (MATLAB®, R2020b)

5 SAATUJEN TULOSTEN ARVIOINTI

Saatuja tuloksia arvioitaessa on hyvä huomioida se, että tämän kandidaatintyön kokeellisen osion tarkoituksena oli havainnollistaa yhtä mahdollista tapaa luoda neuroverkkoluokitin tietyn luokitteluongelman ympärille. Kokeellisen osion tarkoituksena oli esitellä yleisluonteisesti sitä, miten neuroverkkoluokittimen muodostaminen onnistuu hyödyntäen apuna MATLAB®:in lisäosaa, jonka avulla neuroverkkoluokittimia saadaan muodostettua, opetettua ja arvioitua suoraviivaisesti. Saatujen tuloksien arvioiminen tapahtuu tällöin eri näkökulmasta katsoen, kuin vaikkapa silloin, jos kokeellisen osion tarkoituksena olisi virittää neuroverkkoluokitin äärimmilleen tietyn luokitteluongelman ympärille. Äärimmilleen viritetyn neuroverkkoluokittimen tuloksia arvioitaessa keskityttäisiin puhtaasti mahdollisimman optimaalisiin suorituskkyä kuvaaviin numeerisiin tekijöihin, mutta yleisluonteisuuden vuoksi arviointi tapahtuu tässä työssä enneminkin yleisluonteisesta näkökulmasta pohtien, sopivatko keinotekoiset neuroverkot luokitteluongelmien ratkaisumenetelmiksi.

Kuvassa 14 esitettiin testausaineiston onnistuneiden ja epäonnistuneiden luokittelujen osuudet, kun piilokerroksessa olevien neuronien lukumäärää muutettiin. Kuvan pylväsdigrammeja tutkimalla voi huomata sen, että suurin mahdollinen onnistuneiden luokittelujen osuus saadaan, kun piilokerroksessa on 600 neuronina. Käytettäessä piilokerrosta, jossa on 600 neuronina, onnistuneiden luokittelujen osuus on 92,2 % ja vastaavasti epäonnistuneiden luokittelujen osuus on 7,8 %. Tutkittaessa, millä määrällä piilokerroksen neuroneja saavutetaan heikoin tulos, voidaan huomata, että heikon tulos saavutetaan käytettäessä 20 neuronina. Käytettäessä 20 neuronina piilokerroksessa onnistuneiden luokittelujen osuus oli 88,1 % ja epäonnistuneiden 11,9 %. Prosenttiyksiköissä mitattuna parhaan ja heikoimman tuloksen välinen ero onnistuneiden- ja epäonnistuneiden luokitteluiden suhteen on 4,1 prosenttiyksikköä.

Testausaineistoilla saatujen tuloksien oletetaan kertovan mallin hyvyydestä ja yleistyskyvystä ja täten voidaan varauksin olettaa, että onnistunein testausaineistolla saatu tulos voisi korreloida myös opetusaineistoon siinä mielessä, että 600 piilokerroksen neuronilla saavutettaisiin onnistunein tulos myös opetusaineiston suhteen. Opetusaineiston avulla muodostettua neuroverkkoluokitinta voidaan arvioida opetusaineistosta peräisin olevien CE- ja %E-lukujen avulla. Kuvasta 12 pystyi

huomaamaan sen, että opetettaessa neuroverkkoa vaihdellen piilokerroksessa olevien neuronien lukumäärää opetusaineiston CE-luku ja %E-luku vaihtelivat. Sillä otannalla, jolla piilokerroksen neuronien lukumäärää vaihdeltiin, voitiin havaita, että CE-luvun kasvaessa %E-luku pienenee. Tämän tiedon pohjalta voidaan todeta se, että virheiden yhteisvaikutus on matala, jos valitaan esimerkiksi 900 tai 600 neuronia piilokerrokseen. 900 piilokerroksen neuronilla CE-luvuksi saatiin kahden desimaalin tarkkuudella 4,82 ja %E-luvuksi puolestaan 1,14, kun taas 600 neuronilla CE-luvuksi saatiin 4,35 ja %E-luvuksi puolestaan 1,98.

Saatuja tuloksia voidaan tämän työn yhteydessä pitää riittävinä, koska kokeellisen osion tavoitteena oli esitellä yleisluonteisesti, miten neuroverkkoluokittimen saa luotua suoraviivaisesti käyttämällä apuna MATLAB®:in tarjoamaa lisäosaa neuroverkkoluokittimen luomiseen. Kuudellasadalla piilokerroksen neuronilla saavutettiin 92,2 % luokittelutarkkuus, joka tarkoittaa sitä, että neuroverkko kykeni luokittelemaan opetusaineiston 42 000 tapauksesta oikein 38 724 tapausta. Taitavimmat data-analytiikkaa harrastavat henkilöt ovat päässeet Mnist-tietokannan avulla yli 99 % luokittelutarkkuuteen. Saavutettavaan luokittelutarkkuuteen vaikuttaa kuitenkin monet tekijät, kuten esimerkiksi aiempi kokemus data-analytiikasta, aineiston validoinnin ja esikäsitteilyn määrä, keinotekoisien neuroverkon piilokerrosten lukumäärä sekä käytettävä ohjelmisto, jolla neuroverkkoluokittimen muodostaminen suoritetaan. Neuroverkkoluokitin voidaan muodostaa esimerkiksi käyttämällä muodostuksen apuna ohjelmointikielenä C++, R-kieltä tai Pythonia. Data-analytiikkaa ja ohjelmointia vähemmän harjoittaneelle henkilölle MATLAB®:in tarjoama lisäosa neuroverkkoluokittimen muodostamiseen on varsin riittävä ensimmäisen kosketuspinnan luomiseksi keinotekoisien neuroverkkojen ja luokitteluongelmien yhteyteen.

Keinotekoisien neuroverkkojen käytettävyydestä erilaisten luokitteluongelmien parissa on monipuolista näyttöä. Teoriaosiossa esiteltiin muutamia keinotekoisia neuroverkkoja hyödyntäviä luokittelukohteita ja kokeellisessa osiossa esiteltiin neuroverkkoluokittimen muodostaminen, jonka ansiosta keinotekoinen neuroverkko oppi luokittelemaan hetkessä 42 000 tapauksen joukosta 92,2 % luokittelutarkkuudella oikeita tuloksia. Nämä ovat helposti annettavia ja ymmärrettäviä esimerkkejä keinotekoisien neuroverkkojen toimivuudesta luokitteluongelmien ympärillä. Internet on täynnä keinotekoisia neuroverkkoja käsitteleviä tutkimuksia, joista voi halutessaan tutkia lisää, miksi keinotekoisia neuroverkkoja suositaan erilaisissa luokitteluongelmissa.

6 YHTEENVETO

Tiedonlouhinnan harjoittamisen tarve on kasvanut merkittävästi, koska erilaisten kerättävien datojen määrä on digitalisaation ja tietotekniikan kehittymisen myötä kasvanut räjähdysmäisesti. Yritysten on ymmärrettävä data-analytiikan rooli kilpailuilla markkinoilla, joilla yritykset pyrkivät erottumaan kilpailijoistaan korostaen oman palvelunsa tai tuotteen erinomaisuutta. Yrityksillä on rajalliset resurssit, jolloin tuottavuuden maksimointi on ensiarvoisen tärkeää. Pystytäänkö esimerkiksi taloudellisia kustannuksia alentamaan? Pystytäänkö raaka-aineiden hävikkiä pienentämään? Pystytäänkö energiatehokkuutta nostamaan? Kaikkiin edellä mainittuihin kysymyksiin löytyvä vastaus voi hyvinkin piillä yrityksen varastoimassa datassa. Tutkimalla kerättyä dataa esimerkiksi luokittelua, regressiota tai klusterointia apuna käyttäen voidaan mahdollisesti löytää sellaista arvokasta informaatiota, joka parhaimmillaan paljastaa yrityksen tuottavuutta heikentävän niin sanotun pullonkaulan. Tieto esimerkiksi tuotantoa rajoittavasta pullonkaulasta olisi yritykselle äärimmäisen tärkeä. Tiedon merkityksen voi tiivistää vanhan sanonnan mukaan. Englantilainen filosofi ja yleisnero Francis Bacon totesi uuden ajan alkupuolella, tieto on valtaa.

Keinotekoiset neuroverkot ovat yksinkertaisista prosessointielementeistä eli neuroneista koostuvia moniulotteisia tietoa käsitteleviä prosessoreita, jotka kykenevät prosessointikykynsä puolesta käsittelemään saamaansa aineistoa tehokkaasti. Niiden prosessointikyky perustuu siihen, kuinka kvalitatiivisesti sekä kvantitatiivisesti laadukkaalla aineistolla niitä päästään opettamaan. Opetettaessa keinotekoisia neuroverkkoja voidaan hyödyntää backpropagation-algoritmia, jota käytettäessä ollaan eräänlaisessa vuorovaikutuksessa keinotekoisien neuroverkon kanssa annettaessa sille jatkuvasti palautetta sen tuottamien luokittelujen hyvydestä. Opetukseen käytettävän aineiston valinnalla voidaan vaikuttaa siihen, millaiseen käyttötarkoitukseen keinotekoisia neuroverkkoja opetetaan. Massiivisen prosessointikykynsä puolesta keinotekoisia neuroverkkoja hyödynnetään laajalti erilaisten luokitteluongelmien yhteydessä, sillä erilaisten identifiointia vaativien luokitteluongelmien yhteydessä keinotekoisien neuroverkkojen voidaan sanoa olevan omimmillaan. Tyypillisiä identifiointia vaativia luokitteluongelmia ovat esimerkiksi erilaiset kuvan, -puheen- ja tekstintunnistukseen liittyvät luokittelutilanteet.

LÄHDELUETTELO

Garcia S., Herrera F., Luengo J., 2015, Data Preprocessing in Data Mining, Springer, 320 s., ISBN 987-3-319-10246-7. Saatavissa: <http://users.ece.utexas.edu/~ethomaz/courses/dm/papers/data-preprocessing-book.pdf>. [Viitattu 15.1.2021]

Gurney K., 2004, an Introduction to Neural Networks, Routledge, 317 s., ISBN 0-203-45151-1. Saatavissa: https://www.inf.ed.ac.uk/teaching/courses/nlu/assets/reading/Gurney_et_al.pdf [Viitattu 21.2.2021]

Haykin S., 2009, Neural Networks and Learning Machines, 3. painos, Pearson Education Inc., ISBN 978-0-13-147139-9.

North M., 2012, Data Mining for the Masses, Global Text Project, ISBN 978-0-61-156843-76 [Viitattu 19.2.2021]

Shearer C., 2000, The CRISP-DM Model: The New Blueprint for Data Mining, [verkkodokumentti]. Saatavissa: <https://mineracaodedados.files.wordpress.com/2012/04/the-crisp-dm-model-the-new-blueprint-for-data-mining-shearer-colin.pdf>

Starmer J., 2020, “Neural Networks Pt. 1: Inside the Black Box”, StatQuest with Josh Starmer, YouTube-kanava, [YouTube-video]. Saatavissa: <https://www.youtube.com/watch?v=CqOfi41LfDw> [Viitattu 14.2.2021]

Thomas A., An Introduction to neural networks for beginners, [verkkodokumentti]. Saatavissa: <https://adventuresinmachinelearning.com/wp-content/uploads/2017/07/An-introduction-to-neural-networks-for-beginners.pdf> [Viitattu 21.2.2021].

Tuominen H., 2017, Johdatus tekoälyn taustalla olevaan matematiikkaan -kurssi, [verkkodokumentti]. Saatavissa: <https://tim.jyu.fi/view/143092#DKUvbnUuGytQ> [Viitattu 21.2.2021].