jamk.fi

Datan laadun merkitys koneoppimisessa

Panu Partanen

Opinnäytetyö
Toukokuu 2019
Tekniikan ja liikenteen ala
Insinööri (AMK), Tieto- ja viestintätekniikan tutkinto-ohjelma
Ohjelmistotekniikka

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

jamk.fi

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Partanen, Panu	Opinnäytetyö, AMK	Toukokuu 2019
	Sivumäärä 41	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty: x

Työn nimi

Datan laadun merkitys koneoppimisessa

Tutkinto-ohjelma

Tieto- ja viestintätekniikka

Työn ohjaaja(t)

Mika Rantonen, Juha Saarisilta

Toimeksiantaja(t)

Collapick Company Oy

Tiivistelmä

Opinnäytetyössä käsiteltiin koneoppimisen teoriaa ja datan laadun tärkeyttä neuroverkkojen koulutusprosessissa. Tehtävänä oli selvittää asiakkaan työntekijöiden tehtävän kesto käyttämällä koneoppimisenmenetelmiä ja hyödyntämällä yrityksen keräämää tietoa työntekijöiden työtehtävistä.

Tavoitteena oli toteuttaa prototyyppi, jonka tarkoituksena oli ennustaa tulevien tehtävien keston pituus, jotta asiakkaan yrityksen toiminnan eri vaiheita voitaisiin nopeuttaa ja vähentää syntynyttä hukka-aikaa. Tavoitteena oli myös valita sopiva teknologia, joka soveltuisi prototyypin toteutukseen.

Teoreettisessa viitekehyksessä tutkittiin koneoppimisen eri osa-alueita ja vaiheita. Työssä keskityttiin koneoppimisen oppimistyyleihin, neuroverkkojen rakenteeseen ja eri tyyppeihin sekä datan keräämiseen ja analysointiin.

Toteutuksessa käytiin vaiheittain läpi datan esikäsittely- ja normalisointiprosessi sekä neuroverkon luonti ja kouluttaminen valitulla teknologialla.

Lopputuloksena prototyyppi ei onnistunut ennustamaan tehtävän kestoa riittävän tarkasti, jotta sitä voisi hyödyntää konkreettisesti käytännössä. Koulutetun neuroverkon ulostulo ei vastannut syötettyihin tietoihin, koska prototyypissä hyödynnetty data oli huonolaatuista. Yritykselle saatiin kuitenkin tuotua lisäarvoa selvittämällä datan keräämisen ongelmakohtia ja samalla toteutuksessa saatiin todistettua datan laadun merkitys koneoppimisessa.

Avainsanat (asiasanat)

Tekoäly, Koneoppiminen, Neuroverkko, Big data, Tiedon louhinta, Normalisointi, BrainJs

Muut tiedot (Salassa pidettävät liitteet)

jamk.fi

Description

Author(s)	Type of publication	Date
Partanen, Panu	Bachelor's thesis	May 2019
		Language of publication: Finnish
	Number of pages: 41	Permission for web publication: x

Title of publication

Significance of data quality in machine learning

Degree programme

Information and Communication Techology

Supervisor(s)

Rantonen, Mika; Saarisilta, Juha

Assigned by

Collapick Company Oy

Abstract

The thesis covers the theory of machine learning and the importance of data quality in the training for neural networks. The purpose was to determine the duration of the assignor's employees for completing a task using machine learning methods and the information gathered by the company on the employees' assignments.

The goal was to implement a prototype designed to predict the duration length of the future tasks to speed up the separate phases of the assignor's company operations and thus reduce the wasted time. The goal was also to choose the appropriate technology suitable for the prototype.

The theoretical framework explored the different areas and stages of machine learning. The study focused on the learning styles of machine learning, the structure and different types of neural networks, as well as data collection and analysis.

In the implementation, the data preprocessing and normalization as well as the creation and training of the neural network with the selected technology were progressively reviewed.

As the result, the prototype failed to predict the duration of the assignment accurately enough so it could be used concretely in practice. The output of the trained neural network did not correspond to the input because the quality of the data in the prototype was invalid. However, the assignor's company gained additional value with the identification of the problematic areas of data collection and at the same time, the importance of data quality in machine learning was proven.

Keywords/tags (subjects)

Artificial Intelligence, Machine learning, Neural network, Big data, Data mining, BrainJs

Miscellaneous (Confidential information)

Sisältö

Kuviot

Taulukot

Lyhenteet

AI	Artificial Intelligence
CPU	Central processing unit
GPU	Graphics processing unit
IoT	Internet of Things
UI	User interface
UX	User experience

1 Johdanto

1.1 Toimeksiantaja

Toimeksiantajana toimi Collapick Company Oy, jonka toimipisteet sijaitsevat Joensuussa ja Tampereella. Collapickin henkilökuntaan kuuluu kymmenkunta työntekijää. Collapick tarjoaa monipuolisia digiratkaisuja erilaisiin käyttötarkoituksiin. Palveluihin kuuluvat muun muassa mobiilisovelluksien kehittäminen, web-kehitys, UI/UX-suunnittelu ja teollisuuden digitalisoiminen. Teollisuuden digitalisoimiseen kuuluu muun muassa Ponniste-palvelu, joka on tuotannon viestintä ja visualisointijärjestelmä ja se voidaan räätälöidä asiakkaan tarpeen mukaan. Sen avulla vähennetään hukkatyötä ja autetaan työntekijöitä keskittymään arvoa tuottavaan työhön. Collapickin yksi tavoitteista on nostattaa ihmisen tekemän työn arvoa digitalisoituvassa maailmassa. (Collapick 2019.)

Toimeksianto toteutettiin yhteystyössä toimeksiantajan asiakkaan kanssa. Asiakas tarjosi opinnäytetyön toteutukseen tarvittavan datan, jotta toteutus olisi mahdollista toteuttaa vastaamaan reaalimaailman esimerkkiä.

1.2 Tavoitteet ja tehtävät

Tavoitteena opinnäytetyössä oli perehtyä koneoppimisen käytäntöihin ja soveltaa niitä toteuttamalla prototyyppi toimeksiantajalle. Ensisijainen tehtävä oli tutustua ajankohtaisiin koneoppimisen sovelluskehyksiin ja valita niistä yksi, joka soveltuisi opinnäytetyön prototyypin toteutukseen.

Päätavoitteena opinnäytetyössä oli ennakoida työntekijän tehtävän kesto käyttäen apuna koneoppimisenmenetelmiä ja hyödyntäen työntekijöistä kerättyä dataa. Tavoitteena oli saada selville tehtävän valmistuksen ajankohta, jotta seuraavaan tehtävään voitaisiin toimittaa tarvittavat resurssit ennakoivasti.

Prototyypin toteutuksella pystyttäisiin vähentämään tuotannon eri vaiheiden välille syntynyttä hukka-aikaa. Hukka-ajan eliminoinnilla pystytään vähentämään työntekijöiden hukkatyötä ja auttamaan keskittymään arvoa tuottavaan työhön sekä parantamaan yrityksen tuotannon laatua.

Tavoitteena oli myös pohtia, kuinka prototyyppiä pystyisi jatkokehittämään ja hyödyntämään paremmin yrityksen tuotannossa. Lisäksi tutkia erilaisia koneoppimisen käyttötapauksia teollisuuden yrityksissä ja kuinka niitä voitaisiin hyödyntää tuotannossa.

2 Koneoppimisen teoriaa

2.1 Tekoäly

Historia tekoälyn kehitykselle ei ole ollut suoraviivaista. Amerikkalainen tietotekniikan tutkija John McCarthy, joka on laajalti tunnustettu yhdeksi tekoälyn perustajista, järjesti vuonna 1956 Darthmouthin konferenssin, jossa termiä AI (tekoäly) käsiteltiin ensimmäistä kertaa. Tutkijakeskukset yleistyivät Yhdysvalloissa tutkiakseen tekoälyn mahdollisuuksia, ja tutkijat Allen Newell ja Herbert Simon olivat suuressa osassa edistämässä tekoälyä tietotekniikan alana, joka voisi mullistaa maailmaa. (Ray 2018.)

Kuitenkin suuresta rahoituksesta ja useiden vuosikymmenien ponnistelusta huolimatta tietotekniikan tutkijat kokivat suuria vaikeuksia kehittää älykkyyttä koneisiin. Tietokoneet eivät olleet kehittyneet tarpeeksi, jotta olisi ollut mahdollista prosessoida suurta määrää dataa, ja seurauksena siitä valtiot ja yritykset menettivät toivon tekoälyn kehitykseen. Tästä aiheutui ”AI Winters”, jolloin tutkijat kärsivät akuuttia rahoituksen puutetta tekoälyn tutkimuksiin. Kyseinen ajanjakso kesti 1970-luvun puolivälistä 1990-luvun puoliväliin saakka. (Ray 2018.)

Yritykset kiinnostuivat jälleen tekoälyn kehityksestä 1990-luvun loppupuolella ja samoihin aikoihin Japanin hallitus julkisti suunnitelmat kehittää viidennen sukupolven tietokoneen koneoppimisen edistämiseksi. Tekoälyn intoilijat uskoivat, että tietokoneet pystyisivät analysoimaan kuvia, kääntämään kieliä ja hoitamaan tyypillisiä toimintoja ihmisille. (Ray 2018.)

Tekoälyn rahoitus kuitenkin kärsi 2000-luvun alussa tietotekniikan taloudellisesta yliarvostuksesta ja siitä seuranneesta romahduksesta, jota kutsutaan IT-kuplaksi (dotcom bubble). Kuitenkin kehitys jatkui eteenpäin tietokonelaitteiden parannusten ansiosta. Yritykset ja hallitukset jatkoivat onnistuneesti kehitystä alatoimialueilla. Yritykset osasivat hyödyntää tietokoneiden kehittynyttä prosessointitehoa ja tallennuskapasiteettia, joka mahdollisti yritysten tallentaa suuria määriä tietoja ensimmäistä kertaa. (Ray 2018.)

Amazon, Google, Baidu ja muut teknologiayritykset hyödynsivät tekoälyratkaisuja valtavaan kaupalliseen etuunsa viimeisen 15 vuoden aikana. Ensisijaisesti kehitys lähti käyttäjätietojen prosessoinnista, jotta ymmärrettäisiin kuluttajien käyttäytymistä. Kyseiset yritykset ovat kuitenkin jatkaneet ja laajentaneet kehitystä konenäköön, kielten käsittelyyn ja monien muiden tekoälysovellusten pariin. Seurauksena tästä tekoälyratkaisuja on sisälletty moniin tämän päivän verkkopalveluihin. (Ray 2018.)

Erilaisia tekoälyratkaisuja hyödynnetään monilla eri aloilla ja eri tarkoituksiin tänä päivänä. Tyypillisiä käyttötapauksia on havainnollistettu taulukossa 1. (Marr 2016.)

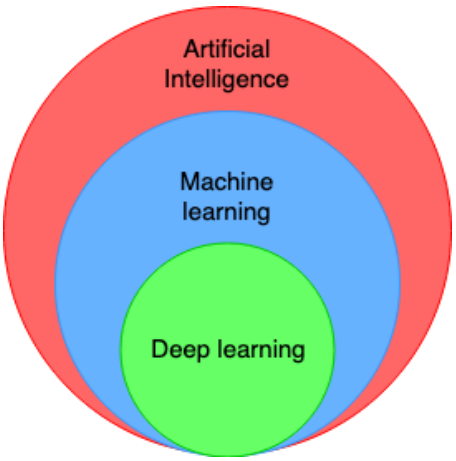
Taulukko 1. Tyypilliset tekoälyn käyttötapaukset

Käyttötapaus	**Kuvaus**
Tietoturva	Haittaohjelmien ja haavoittuvuuksien tunnistaminen
Turvallisuus	Turvatarkastukset ja hätätilanteiden tunnistaminen
Taloudellinen kaupankäynti	Osakemarkkinoiden ennustaminen
Terveydenhuolto	Sairauksien riskitekijöiden tunnistaminen
Markkinoinnin kohdistaminen	Tuotteiden ja palveluiden kohdistaminen asiakkaille yksilöllisesti
Petosten tunnistaminen	Laiton liiketoiminta
Suositukset	Seurataan asiakkaan toimintaa, jonka pohjalta suositellaan yksilöllisesti palvelun tarjontaa
Kuvan tunnistus	Ympäristön havainnointi älyautossa
Luonnollinen kielenkäsittely	Puheen tunnistus ja kielikäännökset

Tekoälyn toiminnot pyrkivät jäljittelemään älykkyyttä vaativia toimintoja, jotka ovat usein tyypillisiä ihmisille. Vaativia toimintoja voivat olla muun muassa puheentunnistus, päätöksenteko, käännökset eri kielien välillä ja visuaaliset havainnot. (Artificial Intelligence vs. Machine Learning vs. Deep Learning n.d.)

Tekoäly voidaan jakaa kahteen eri ryhmään: kapeaan ja vahvaan tekoälyyn. Kapeaan tekoälyyn sisältyvät kaikki modernit tekoälyratkaisut, kun taas vahva tekoäly kykenisi selviytymään kaikista yleistoiminnoista mistä ihminenkin. Systeemi, joka omaisi vahvan tekoälyn, muistuttaisi paljon ihmistä. Systeemillä olisi kognitiivisia taitoja ja yleinen kokemuksellinen ymmärrys sen ympäristöstä. (Trask 2018.)

Tekoälyn (artificial intelligence), koneoppimisen (machine learning) ja syväoppimisen (deep learning) suhdetta toisiinsa voidaan verrata maatuskaan eli joukkoon venäläisiä nukkeja, jotka ovat sisäkkäin toisiaan. Päällimmäinen kerros olisi tekoäly, esimerkiksi mikä tahansa tietokoneohjelma, joka pystyisi tekemään jotain älykästä. Seuraava sisäkkäinen kerros olisi koneoppiminen, joka on tekoälyn yksi alakategorioista, jossa tietokoneohjelma oppii kerätystä datasta ilman ihmisen vuorovaikutusta. Kuitenkaan kaikki tekoäly ei ole koneoppimista, mutta kaikki koneoppiminen on osa tekoälyä. Viimeisin sisäkkäinen kerros olisi syväoppiminen, joka on alakategoria koneoppimiselle ja se mahdollistaa koneita oppimaan esimerkkien avulla. Alakategorioita on havainnollistettu kuvioissa 1. (Artificial Intelligence vs. Machine Learning vs. Deep Learning. n.d.)



Kuvio 1. Tekoäly, koneoppiminen ja syväoppiminen

2.2 Koneoppiminen

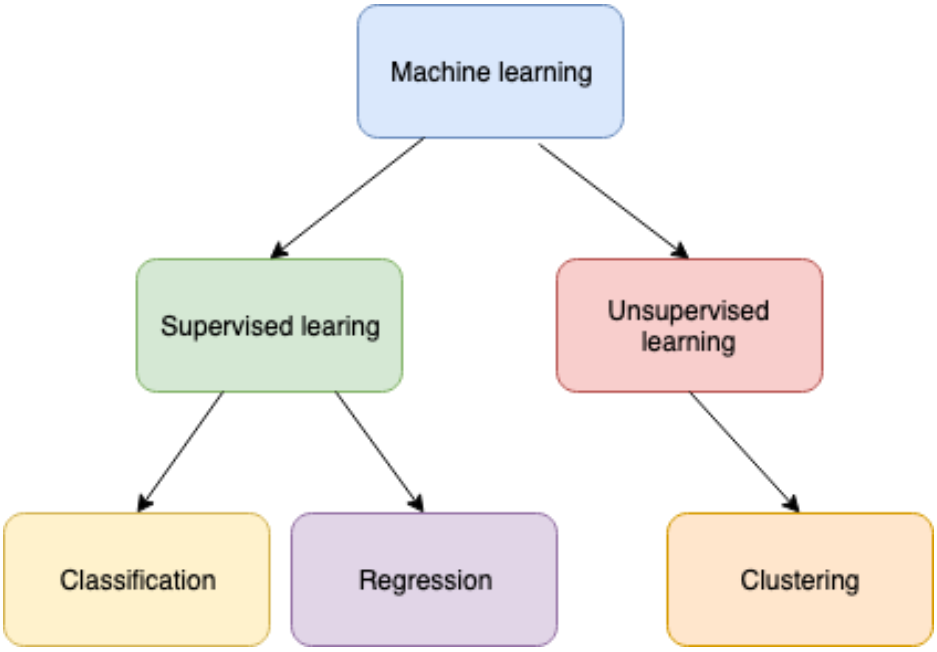
Koneoppimisen yksi merkkipaaluista on vuoden 1950 ”Turingin testi”, jonka Alan Turing kehitti selvittääkseen, onko tietokoneella todellinen älykkyys. Testin suorittamiseksi tietokoneen on kyettävä huijaamaan ihmistä, että sekin on ihminen. Kuitenkin ensimmäisen tietokoneohjelman, joka pystyi oppimaan ajon aikana, kehitti Arthur Samuel. Tietokoneohjelma oli peli nimeltä Tammi. Se oppi tunnistamaan, mitkä liikkeet muodostivat voittostrategioita, ja sisälsi kyseiset muutoksen ohjelmaansa ajon aikana. eli tietokoneohjelma kehittyi koko ajan mitä, enemmän sitä pelattiin. (Mayo, Punchihewa, Emile & Morrison 2018.)

Koneoppiminen on yksi tekoälyn osa-alue. Se pohjautuu ajatukseen, jossa järjestelmät voivat oppia kerätystä datasta, identifioida toistuvia kuvioita ja tehdä päätöksiä ilman, että ihminen joutuisi olemaan vuorovaikutuksessa järjestelmään. Koneoppiminen on tieteellinen tutkimus algoritmeista ja tilastollisista malleista, joiden avulla ohjelmistosovellukset voivat ennustaa tuloksia tarkasti ilman, että niitä olisi erityisesti ohjelmoitu. Perusperiaate on rakentaa algoritmeja, jotka pystyvät vastaanottamaan tietoa ja käyttämään tilastollisia analyyseja ennustamaan päivittäen ulostulotietoa sitä mukaan, kun uutta tietoa on saatavilla. (Machine Learning – What it is and why it matters n.d.)

Tyypillisesti koneoppiminen voidaan kategorisoida oppimistyylin perusteella ohjattuun oppimiseen (supervised learning), ohjaamattomaan oppimiseen (unsupervised learning) sekä vahvistettuun oppimiseen (reinforcement learning). Oppimistyylit poikkeavat toisistaan lähestymistavassaan ongelmien ratkaisussa, toiminnoissaan ja tiedon tyypissä. (What is Machine Learning? A definition N.d.)

Ohjattu oppiminen perustuu koneen opettamiseen luokitellun tiedon avulla. Tavoitteena on, että kone pystyisi tekemään luokittelun samankaltaisille tiedoille. Tuloksen luonteen perusteella ohjattu oppiminen voidaan jakaa kahteen eri luokkaan: luokittelu (classification) ja regressio (regression). Kyseessä on luokittelu, jos tulosta pystyy kategorisoimaan, esimerkiksi eläinten tunnistaminen kuvista. Regressiota on, jos tulos on jatkuvaa, esimerkiksi jos kuvista halutaan tunnistaa eläimen korkeus. Luokitellun koulutusaineiston analysoinnin jälkeen oppimisalgoritmi pystyy tuottamaan tuloksen toimintojensa perusteella. Tulosta pystytään vertaamaan testiaineiston tuloksiin ja säätämään oppimisalgoritmia vastamaan haluttuihin tuloksiin. (What is Machine Learning? A definition N.d.)

Ohjaamatonta oppimista on, kun koulutusaineisto ei ole luokiteltu tai nimetty. Tavoitteena koneella on tunnistaa koulutusaineistosta riippuvaisuuksia ja samankaltaisuuksia eri tietojen välille. Tarkoituksena on pyrkiä ryhmittelemään (clustering) tietoa sen ominaisuuksien perusteella. (What is Machine Learning? A definition N.d.) Tyypilliset oppimistyylit on havainnollistettu kuviossa 2.



Kuvio 2. Koneoppimisen tyypilliset oppimistyylit

Lisäksi on olemassa puoliohjattua oppimista (semi-supervised learning), joka jää ohjatun ja ohjaamattoman oppimisen välimaastoon. Siinä käytetään sekä luokiteltua että luokittelematonta tietojoukkoa koulutukseen. Tyypillisesti tietojoukot sisältävät pienen määrän luokiteltua tietoa ja suuren määrän luokittelematonta tietoa. Koneet, jotka käyttävät kyseistä oppimismenetelmää, voivat parantaa huomattavasti oppimistarkkuutta. Tyypillisesti puoliohjattu oppiminen valitaan, jos luokiteltu tieto vaatii huomattavan paljon ammattiosaamista ja asianmukaisia resursseja. (What is Machine Learning? A definition N.d.)

Viimeisenä oppimistyylinä on vahvistettu oppiminen (reinforcement learning). Se on oppimismenetelmä, jossa kone on vuorovaikutuksessa sen ympäristön kanssa tuottamalla toimintoja ja etsimällä positiivista tai negatiivista palautetta. Palautteen antaminen koneelle, joka käyttää vahvistettua oppimista, on välttämätöntä, jotta algoritmi voi selvittää, mikä toiminto on paras tulosten kannalta. (What is Machine Learning? A definition n.d.)

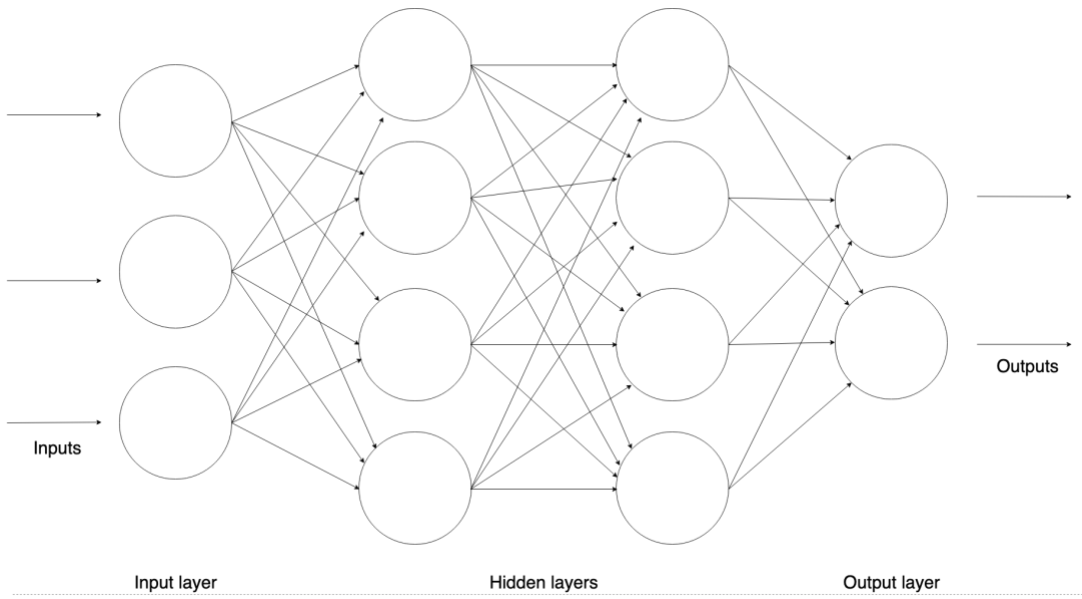
2.3 Neuroverkko

Ensimmäinen tapaus neuroverkoista oli vuonna 1943, kun neurofysiologisti Warren McCulloch ja matemaatikko Walter Pitts kirjoittivat artikkelin neuroneista ja niiden toiminnoista. McCulloch ja Pitts mallinsivat yksinkertaisen neuroverkon sähköpiirillä. (Mayo, Punchihewa, Emile & Morrison 2018.)

Neuroverkko on joukko algoritmeja, jonka rakenne muistuttaa aivojen hermosolujen verkottunutta rakennetta. Neuroverkko pystyy oppimaan syötetyistä tiedoista ja tunnistamaan kuvioita, luokittelemaan tietoa ja ennustamaan tulevia tapahtumia. (Mayo, Punchihewa, Emile & Morrison 2018.)

Neuroverkko pilkkoo syötetyn tiedon abstrakteihin kerroksiin, joita voidaan kouluttaa suurella määrällä tietoa esimerkiksi tunnistamalla trendejä kuvista samalla lailla kuten ihminenkin. Neuroverkon käyttäytyminen pohjautuu yksittäisen solmujen yhdistämiseen toisiinsa niiden vahvuuksien ja painoarvojen mukaan. Painoarvot pystytään asettamaan oletusarvoksi jo ennen neuroverkon kouluttamista. Kuitenkin koulutuksen aikana painoarvot säädetään automaattisesti riippuen tiedosta ja neuroverkon ominaisuuksista. Painoarvot eivät enää muutu, jos neuroverkko suorittaa halutun toiminnon onnistuneesti. (Mayo, Punchihewa, Emile & Morrison 2018.)

Neuroverkko yhdistää useita rinnakkain toimivia prosessointikerroksia, jotka käyttävät yksinkertaisia solmuja, jotka toimivat rinnakkain. Neuroverkko koostuu syötekerroksesta (input layer), yhdestä tai useammasta piilotetusta kerroksesta (hidden layer) ja ulostulokerroksesta (output layer). Lisäksi jokainen kerros voi sisältää useita solmuja. Havainnollistava kuva neuroverkon rakenteesta on kuviossa 3. (What Is a Neural Network? n.d.)



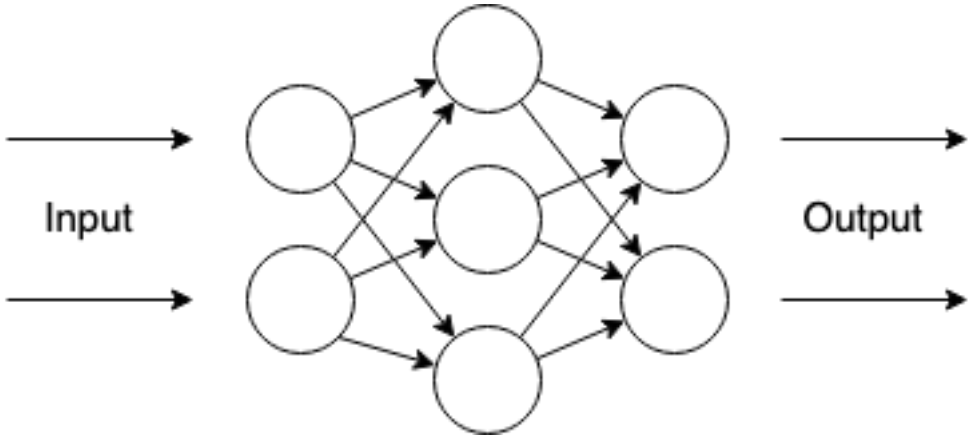
Kuvio 3. Neuroverkon rakenne

Tavoitteena neuroverkon koulutuksessa on minimoida ulostulojen ja tavoitearvojen välinen opetusvirhe eli ulostulon ja tavoitearvon välinen ero. Kouluttaakseen neuroverkkoa on otettava huomioon, että neuroverkkoa ei ylisoviteta (overfitting). Eli jos opetettu malli noudattaa täysin koulutettua tietoa, neuroverkko ei pysty ennustamaan annettujen tietojen ulkopuolelle. Vastakohtana on alisovittaminen, jolloin neuroverkkoa ei ole koulutettu tarpeeksi. (underfitting). Liian vähän koulutettu malli ei kykene yleistämään tietoa. (What Is a Neural Network? n.d.)

Tyypillisiä ratkaisuja yli- ja alisovittamisen ehkäisemiseksi ovat sopiva määrä piilotettuja kerroksia ja piilotettujen kerrosten solmuja, iteraatioden määrä, sekä opetuksen pysäytys. Opetuksen ennen aikaisessa pysäytyksessä syötetieto jaetaan opetus- ja validointidataan. Neuroverkkoa koulutetaan opetusdatalla ja validointidata testataan yleistysvirheellä. Kun yleistysvirhe alkaa kasvamaan tai saavuttaa asetetun tietyn arvon, neuroverkon koulutus pysäytetään. (Katsaus neuroverkkomallintamiseen n.d.)

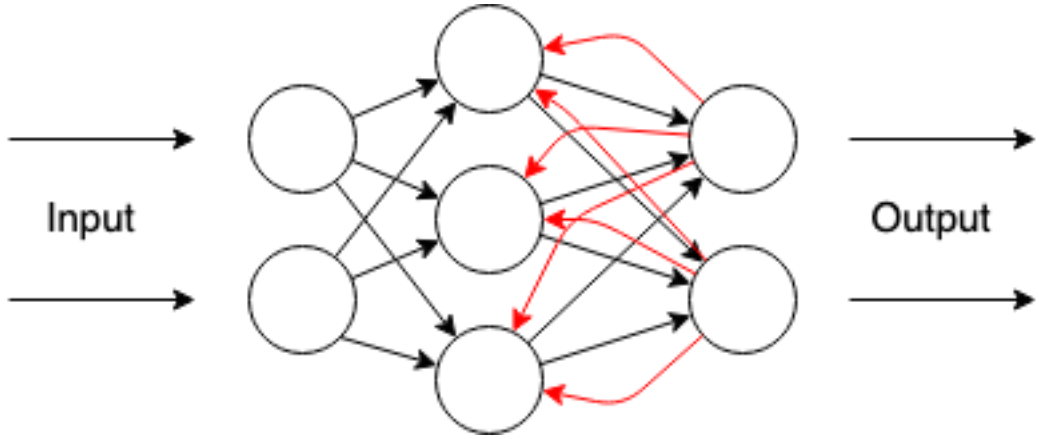
Eri tyyppiset neuroverkot käyttävät erilaisia periaatteita omien sääntöjensä määrittämiseen. Ne soveltuvat omilla vahvuuksillaan erilaisiin tehtäviin paremmin kuin toiset. (Mehta 2019.)

Yksinkertaisin neuroverkkojen tyypeistä on feedforward-neuroverkko, jossa tieto kulkee eri syötesolmujen läpi niin kauan, kunnes se saavuttaa ulostulosolmun. Feedforward-neuroverkossa ei ole vastavirtaa toiseen suuntaan (backpropagation) kuten monimutkaisemmissa neuroverkkotyypeissä, vaan tieto siirtyy ainoastaan yhteen suuntaan. Kuitenkin siinä pystyy olemaan yksi tai useampi piilotettu kerros, jossa lasketaan syötteiden ja niiden painoarvojen summa ulostulolle. Havainnollistava kuva Feedforward-neuroverkosta on kuviossa 4. (Mehta 2019.)



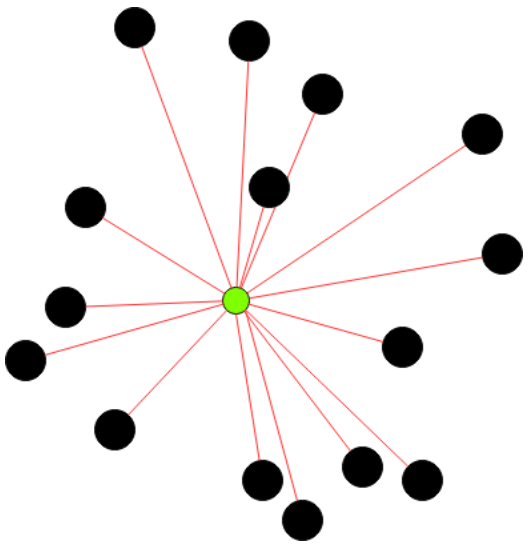
Kuvio 4. Feedforward-neuroverkko

Multilayer perceptron -neuroverkossa on ainakin kolme kerrosta: syötekerros, yksi tai useampi piilotettu kerros ja ulostuloskerros. Kyseinen neuroverkkotyyppi on feedforward tyyppisen neuroverkon luokka, jossa se käyttää ohjatun oppimisen tekniikkaa backpropagation, jossa kukin solmu on kytketty toisiinsa lukuun ottamatta ulostulosolmuja. Backpropagation mahdollistaa lähtövirheen laskemisen ja liikkumisen ulostulokerroksesta piilotettuun kerrokseen, jotta solmujen painoarvoja voidaan säätää siten, että virhearvo laskee. Solmut käyttävät epälineaarista aktivointitoimintoa, jotta neuroverkolla pystyttäisiin erottamaan tietoa, jota ei voida erottaa lineaarisesti. Epälineaariset aktivointitoiminnot mahdollistavat neuroverkon luoda monimutkaisia kartoituksia verkon syötetulojen ja ulostulojen välillä. Ne ovat välttämättömiä monimutkaisten tietojoukkojen oppimisessa, koska tietojoukot ovat epälineaarisia ja niiden ulottuvuus on laaja. Epälineaariset toiminnot mahdollistavat backpropagation-tekniikan käytön, koska niillä on johdannainen toiminto syötetuloihin. Tämän tyyppistä neuroverkkoa käytetään tyypillisesti puheentunnistus- ja kääntämistekniikoissa. Havainnollistava kuva multilayer perceptron -neuroverkosta kuviossa on 5. (Mehta 2019.)



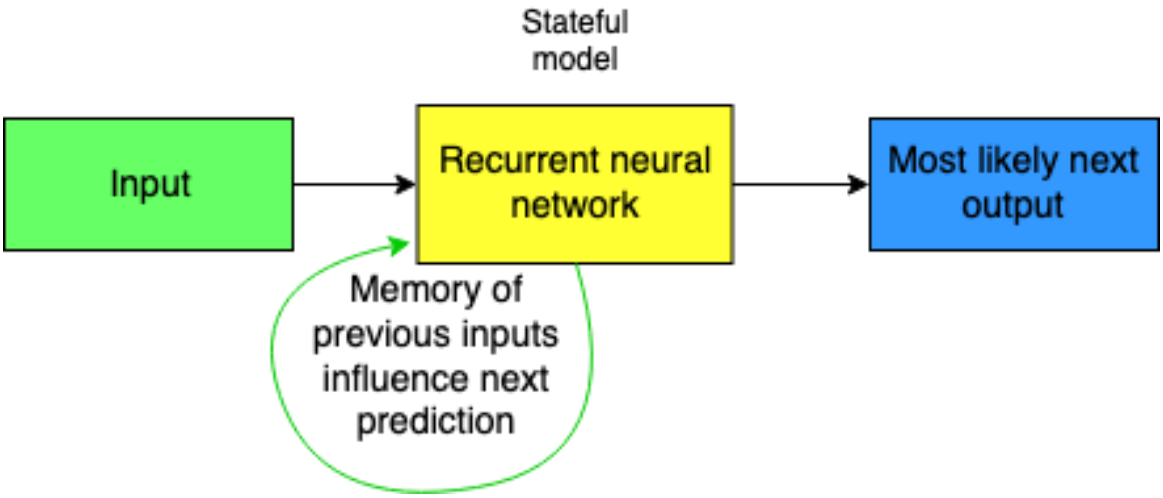
Kuvio 5. Multilayer perceptron -neuroverkko

Radial based -neuroverkko tarkastelee keskustaan nähden etäisyyttä mistä tahansa pisteestä. Neuroverkolla on kaksi kerrosta: sisäkerroksen ominaisuudet yhdistetään säteittäiseen perustoimintoon. Ominaisuuksien ulostulo otetaan huomioon laskettaessa samaa ulostuloa seuraavassa vaiheessa. Tyypillisesti radiaalipohjaista neuroverkkoa käytetään erilaisissa palautusjärjestelmissä, jossa virta voidaan palauttaa mahdollisimman lyhyessä ajassa. Havainnollistava kuva radial based -neuroverkosta on kuviossa 6. (Mehta 2019.)



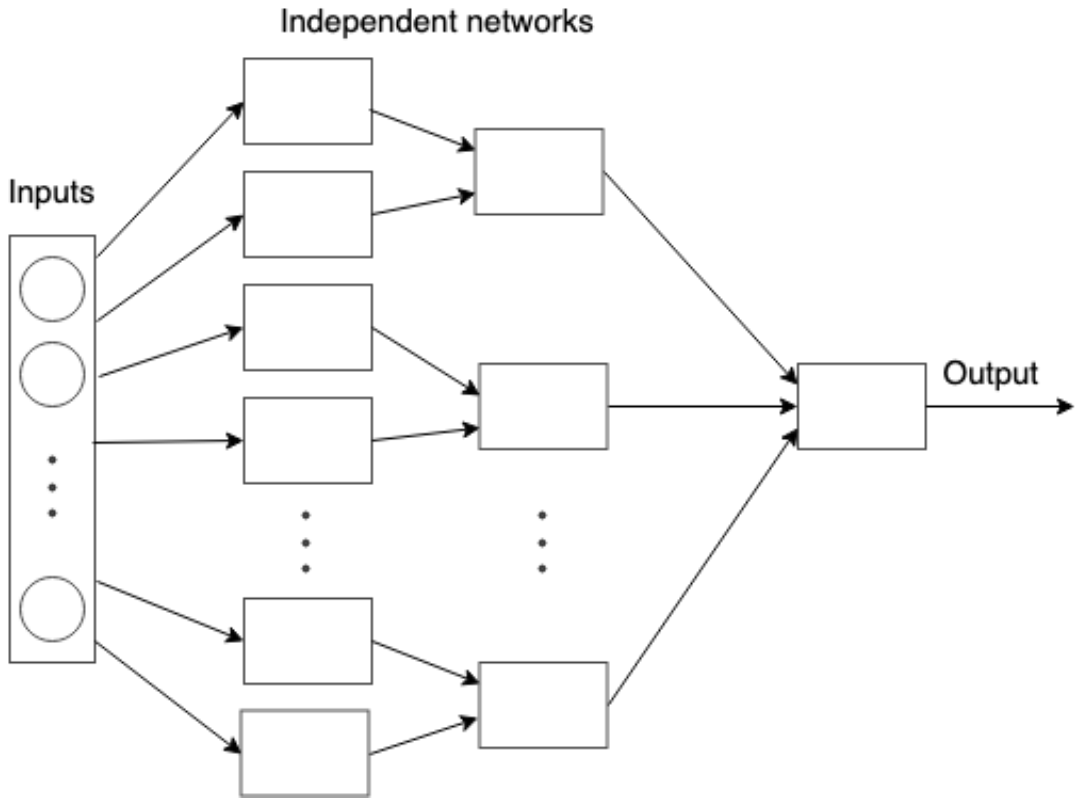
Kuvio 6. Radial based -neuroverkko

Recurrent-neuroverkossa ensimmäinen kerros on samanlainen kuin feedforward-neuroverkossa. Eroavaisuus tulee ilmi seuraavissa kerroksissa. Kullakin kerroksen solmulla on jokaisesta aika-askeleesta toiseen muistettavaa tietoa, jota solmulla oli edellisessä vaiheessa. Eli solmut toimivat myös muistisoluina, kun lasketaan ja suoritetaan operaatioita. Tämä auttaa neuroverkkoa ennustamaan kerroksen lopputulosta, ja jos ennuste on väärä, järjestelmä oppii itsestään ja yrittää pyrkiä tekemään oikean ennusteen backpropagation vaiheen aikana. Recurrent-neuroverkko on erittäin tehokas esimerkiksi tekstin muuttamisessa puheeksi. Havainnollistava kuva recurrent-neuroverkosta kuviossa on 7. (Mehta 2019.)



Kuvio 7. Recurrent-neuroverkko

Modular-neuroverkko koostuu useasta neuroverkosta, jossa neuroverkot saattavat olla useaa eri tyyppiäkin. Neuroverkot toimivat itsenäisesti ja suorittavat erilaisia alitehtäviä. Neuroverkot eivät ole vuorovaikutuksessa toisiinsa, vaan pyrkivät itsenäisesti selvittämään ulostulon. Tällä tavoin monimutkainen ja raskas laskentaprosessi voidaan toteuttaa huomattavasti kevyemmin, kun neuroverkko pilkotaan itsenäisiin komponentteihin. Havainnollistava kuva modular-neuroverkosta on kuviossa 8. (Mehta 2019.)



Kuvio 8. Modular-neuroverkko

2.4 Syväoppiminen

Syväoppiminen on koneoppimisen osa-alue. Termi viittaa neuroverkon piilotettujen kerrosten lukumäärään. Pinnallisessa neuroverkossa (shallow neural network) on ainoastaan yksi piilotettu kerros, kun taas syvässä neuroverkossa (deep neural network) on useita piilotettuja kerroksia. Useat piilotetut kerrokset mahdollistavat neuroverkon oppivan tietojen ominaisuuksia niin sanotussa ominaisuushierarkiassa. Syvät neuroverkot siirtävät syötetietoja useampien laskennallisten toimintojen avulla. Artificial Intelligence vs. Machine Learning vs. Deep Learning. n.d.)

Syväoppiminen mahdollistaa tarkemman tuloksen monimutkaisemmille ja suurille tietojoukoille. Vaikka syväoppiminen tuli ensimmäisen kerran teoriassa esille 1980-luvulla, vasta viime vuosina siitä on tullut hyödyllistä arkipäivään. Syinä siihen ovat, että syväoppiminen vaatii suuren määrän strukturoitua tietoa ja huomattavaa laskentatehoa tietokoneelta. (What is Deep Learning? n.d.)

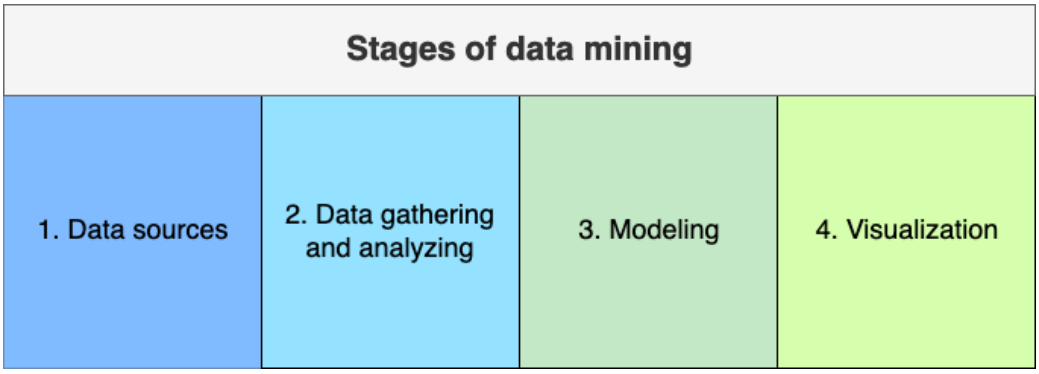
2.5 Tiedon louhinta

Konseptina tiedon louhinta (data mining) on ollut läsnä jo yli vuosisadan, mutta tuli julkisuuteen 1930-luvulla, kun Alan Turing esitteli idean universaalista laitteesta, joka voisi suorittaa laskelmia samalla tavoin kuin nykyaikaiset tietokoneet. Termi tiedon louhinta kuitenkin ilmeni vasta 1990-luvulla. (Data Mining vs. Machine Learning: What´s The Difference 2017.)

Tiedon määrä kasvaa jatkuvasti digitaalisessa maailmassa ja 90 prosenttia tiedosta on pelkästään strukturoimatonta tietoa, joka yleensä esiintyy nimellä big data. Enemmän tietoa ei tarkoita välttämättä, että olisi enemmän tietämystä. Siksi tiedon louhinta on oleellista, jotta strukturoimatonta tiedosta voitaisiin saata uutta tietoa esille, jota voitaisiin hyödyntää muun muassa koneoppimisessa.

Tiedon louhinnan avulla voidaan analysoida piilotettuja yhteneväisyyksiä suuressa määrässä strukturoimatonta tietoa eri näkökulmien mukaan, jotta saataisiin luokiteltua tietoa hyödylliseksi. Analysoitu tieto kerätään ja yritetään hyödyntää esimerkiksi yrityksen liiketoiminnan tulojen lisäämiseksi. (Data Mining What it is and why it matters N.d.)

Tyypillisiä vaiheita tiedon louhinnalle on tietolähteen hankkiminen, missä strukturoimaton tieto ylläpidetään. Useissa tapauksissa yritykset saattavat suoraan siirtää suuren määrän epäolennaista tietoa talteen valmiiksi analysoitavaksi. Tietojen varastoinnin jälkeen tiedoille on annettava käyttömahdollisuus esimerkiksi yritystoiminnan analyytikoille. Tiedon louhintaan valitaan sopiva algoritmi, jolla pystytään tunnistamaan trendejä ja suhteita tietojen välillä. Seuraavaksi vaikeaselkoisesta tiedosta saadaan mahdollisesti luokiteltua tietoa, jota voidaan hyödyntää yrityksen liiketoiminnassa. Lopuksi luokiteltu tieto visualisoidaan helposti ymmärrettäväksi. Havainnollistava kuva tiedon louhinnan vaiheista kuviossa 9. (Data Mining N.d.)

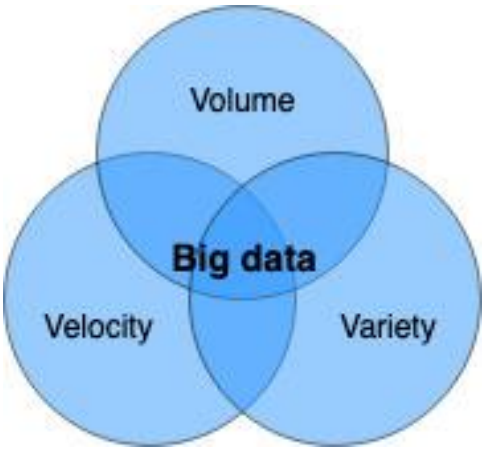


Kuvio 9. Tiedon louhinnan vaiheita

2.6 Big data

Big datan määritelmä voi vaihdella paikoittain, mutta lyhyesti big data on massiivisempaa, vaikeaselkoisempaa, jatkuvasti kasvavaa ja nopealla aikavälillä tulevaa tietoa, joita perinteiset tietojenkäsittelysovellukset eivät kykene hallitsemaan. Kuitenkin big datalla voidaan osoittaa liiketoiminnan ongelmia, joita ei aikaisemmin pystynyt selvittämään. (What is big data? n.d.)

Tyypillisiä piirteitä big datalle on suuri määrä (volume), joka voi olla yrityksillä esimerkiksi sensoridataa, jota luetaan laitteelta. Joillain yrityksillä määrä voi olla kymmeniä teratavuja tai jopa satoja petatavuja tietoa. Toinen tyypillinen piirre on tiedon nopeus (velocity) eli kuinka nopeasti tieto vastaanotetaan sekä mahdollisesti käsitellään. Joissain tapauksissa tietoa ei välttämättä tallenneta ollenkaan kovalevylle, vaan tieto virrataan suoraan muistiin ja arvioidaan reaaliajassa sekä mahdollisesti käytetään koneoppimisenmenetelmiä kysytyn kysymyksen selvittämisessä. Lisäksi tyypillinen piirre big datalle on tiedon valikoima (variety) eli tiedon monipuolisuus. Yleisesti tieto on strukturoitua ja sijaitsee relaatiotietokannassa. Kuitenkin big datan nousun myötä tieto vastaanotetaan strukturoimattomana tai puolistrukturoituna kuten tekstinä, kuvina, äänitteinä ja videoina, jotka vaativat esiprosessoinnin ennen kuin voidaan määritellä tiedon merkitys tai hyödyntää käytännössä. Tyypilliset piirteet on esitetty kuviossa 10. (What is big data? n.d.)



Kuvio 10. Big datan 3 V:tä

Käsitteenä big data itsessään on suhteellisen uusi termi. Alkuperä suurilla tietojoukoilla sijoittuu 1960- ja 1970-luvulle, kun tietojen maailma oli juuri alkamassa ensimmäisten datakeskusten ja relaatiotietokannan kehityksen yhteydessä. Ihmiset alkoivat tajuamaan 2000-luvun alussa, kuinka paljon dataa käyttäjät tuottivat verkkopalveluilla. (What is big data? n.d.)

Samoihin aikoihin kehitettiin avoimen lähdekoodin sovelluskehys Hadoop, joka oli nimenomaisesti tarkoitettu suurien tietojoukkojen tallentamiseen ja analysointiin. Avoimen lähdekoodin sovelluskehysten kehittäminen, kuten aikaisemmin mainittu Hadoop oli olennainen osa big datan kasvussa, koska sovelluskehykset helpotti työskentely suurien tietojoukkojen kanssa ja sitä oli halvempaa tallentaa ja ylläpitää. (What is big data? N.d.)

Viime vuosina big datan määrä on kasvanut moninkertaisesti, koska käyttäjien tuottaman suuren tiedon määrän lisäksi IoT-toteutukset ovat yleistyneet. Tyypillisesti erilaiset toteutukset ovat laitteita, jotka ovat yhteydessä internetiin ja keräävät tietoa käyttäjien toiminnoista ja laitteen suorituskyvystä. Koneoppimisen kehittymisen yhteydessä tietoa on saatu tuotettua lisää. (What is big data? n.d.)

Tyypillinen käyttötapaus big dataan liittyen on koneoppiminen, mikä on yksi olennainen syy, miksi koneoppiminen on ajankohtainen aihe. Enää ei tarvitse ohjelmoida laitteita tai sovelluksia analysoimaan tietojoukkoja vaan laitteita voidaan opettaa. Koneoppimismallien kouluttamisen mahdollistaa big datan saatavuus. Big dataa voidaan hyödyntää myös yrityksen erilaisissa toiminnoissa muun muassa asiakaskokemuksen parantamisessa, virhetilanteiden ennakoimisessa ja tuotannon kehityksessä, joissa yleensä on käytetty koneoppimisenmenetelmiä apuna, koska tietojoukkojen määrä on niin massiivista ja jatkuvasti kasvavaa. (What is big data? n.d.)

Big data tuo mukanaan myös haasteita, suuret tietojoukot tuplaantuvat määrällisesti noin kahden vuoden välein ja seurauksena yritykset kamppailevat pysyäkseen tietojensa tahdissa ja löytääkseen uusia tehokkaita tapoja tallentaa keräämiään tietojaan (Turner 2014). Kuitenkaan tiedon tallentaminen ei ole ainut haaste. Haastattelujen mukaan tietotieteilijät käyttävät 50-80 prosenttia ajastaan tiedon analysointiin ja esikäsittelyyn, ennen kuin sitä voidaan hyödyntää käytännössä (Lohr 2014). Esikäsittelyn ansiosta saadaan puhdasta tietoa eli tietoa mikä on oleellista yritykselle tai asiakkaalle sekä organisoitu tavalla, joka mahdollistaa merkityksellisen analyysin ja hyödyntämisen koneoppimisenmenetelmissä. (What is big data? n.d.)

Lisäksi jatkuvana haasteena on pysyä perässä teknologioiden nopeassa kehityksessä. Tänä päivänä paras tapa käsitellä big dataa näyttäisi olevan kahden eri sovelluskehyksen kombinaatio, Apache Hadoop ja Apache Spark. (What is big data? n.d.)

2.7 Datan laadun merkitys ja esikäsittely

Koneoppimisessa olennainen ja perustuvanlaatuinen totuus on, että ohjelmisto, joka hyödyntää koneoppimisen menetelmiä voi olla vain yhtä hyvä kuin sille syötetty data. Vaikka ohjelmisto olisi suunniteltu ja toteutettu täydelliseksi, se ei muuta sitä totuutta, että ulostulo ohjelmistosta olisi millään tavalla hyödyllistä tai merkittävää jo sille syötetty data on heikkoa laadultaan. Heikkolaatuinen data voi olla täynnä epäpuhtauksia, täynnä korruptiota, päällekkäisyyksiä, poikkeamia ja täynnä epätarkkuuksia. (Machine Learning and Data Quality n.d.)

Monesti ohjelmoijat törmäävät erilaisissa yrityksissä ongelmaan, jossa datan laatu on heikkoa, kun halutaan hyödyntää olemassa olevaa dataa koneoppimisen menetelmillä ja selvittää erityyppisiin ongelmiin vastauksia. Kuitenkin datan laadun selvittämisestä saadaan jo yritykselle ongelmakohtia esille ja sitä kautta tietoa, siitä miten liiketoimintaa pitäisi parantaa, jotta datan hyödyntäminen olisi mahdollista. (Machine Learning and Data Quality n.d.)

Datan laatua voidaan parantaa, muun muassa selvittämällä laadun ongelmatekijät ja sitä kautta suodattamalla epäolennaista tietoa pois. Useasti yrityksissä tietoa saattaa tulla useasta eri lähteestä ja vielä tyypillisesti kolmannen osapuolen lähteistä, jolloin tieto ei ole johdonmukaista. Tiedoissa saattaa ilmetä runsaasti poikkeavaisuuksia ja päällekkäisyyksiä, jotka ovat haitaksi ohjelmiston opetusprosessille. Tällöin on olennaista validoida ja puhdistaa tieto yhteiseen muotoon, jotta tietojen välillä olisi yhdenmukaisuutta. Huonolaatuista tietoa, ei kuitenkaan välttämättä voi aina pelastaa erilaisilla puhdistus- ja suodatusmenetelmillä. Esimerkiksi jos yrityksen keräämä tieto on kerätty työntekijöiltä tai asiakkailta epäjohdonmukaisesti erilaisista lomakkeista, joihin henkilö on itse päässyt syöttämään tietoa haluamallaan tavallaan.

Tarpeeksi kattavalla validoinnilla ja ohjeistuksella lomakkeista saadaan ulos yhdenmukaista tietoa, mutta joissain tapauksissa yritykset joutuvat muuttamaan täysin toimintamalliaan tiedonkeräykseen, jotta laadukasta dataa olisi saatavilla. (Machine Learning and Data Quality n.d.)

Normalisointi on tekniikka, jota käytetään usein osana koneoppimisen datan esikäsittelyä. Tavoitteena normalisoinnissa on muuttaa tietojoukkojen arvot yhteiseen mittakaavaan vääristämättä arvojen vaihteluvälejä. Kuitenkin normalisointia vaaditaan vain ainoastaan silloin, kun ominaisuuksilla on erilaisia arvoalueita. Normalisointi on vektorin jakamisprosessi sen pituuden mukaan, jonka lopputuloksena annettu data saa arvon välillä 0 ja 1. Kuitenkin normalisointi altistuu herkästi poikkeamille, joten on oleellista suodattaa ja validoida data puhtaaksi ennen normalisointia. (Almaliki 2019.)

3 Toteutus

3.1 Teknologian valinta

Opinnäytetyöhön koneoppimisen teknologiaksi valittiin avoimen lähdekoodin kirjasto nimeltä BrainJs. BrainJs on Javascript pohjainen kirjasto, joka yksinkertaistaa hermoverkkojen määrittely-, koulutus ja käyttöprosessia. Yksinkertaisuuden ansioista kirjasto mahdollistaa neuroverkon rakentamisen ja toteuttamisen ilman suurta taustatietoa eli opetuskäyrä kirjaston käyttöön saattaa olla matalampi kuin muissa koneoppimisen kirjastoissa. Dokumentointi BrainJs kirjastossa on selkeää ja ytimekästä, mutta tiedon monipuolisuus kuitenkin oli vain pintaraapaisu koneoppimiseen ja neuroverkon toteutukseen.

Tehokkuuden kannalta BrainJs kirjasto ei välttämättä ole paras vaihtoehto, koska se on kirjoitettu Javascript ohjelmointikielellä, jolloin sitä ei voida kääntää suoraan konekieleen, eikä käyttämään suoraan CPU:ta ja GPU:ta. Lisäksi Javascriptissa on erittäin minimaalinen hyöty koneoppimisen kehittäjille, koska se ei sisällä valmiina höydyllisiä kirjastoja koneoppimiseen tai tiedonkäsittelyyn. Tämän takia tyypillisesti koneoppimisen ohjelmointikielenä käytetään Pythonia, koska se kääntyy suoraan konekieleksi ja pystyy käyttämään suoraan CPU:ta ja GPU:ta. Hyvänä lisänä Pythonissa on kirjastojen saatavuus kuten Numpy, Pandas, Searborn ja Keras, jotka ovat käteviä koneoppimisen kehittämisessä. Lisäksi Pythonissa on huomattavasti suurempi yhteisö koneoppimiseen liittyen eli erilaisiin ongelmatilanteisiin löytyy helpommin vastauksia ja apua kuin Javascript yhteisössä. Huomiona kuitenkin koneoppiminen Javascript maailmassa on ottanut tuulta alleen viime aikoina ja suosion suunta on kasvamassa ylöspäin.

BrainJs soveltui kuitenkin opinnäytetyön prototyypin toteuttamiseen, sillä ongelma mitä haluttiin ratkaista, oli yksinkertainen. Yksinkertaisuuden ansiosta neuroverkon kehittämiseen pääsi nopeasti sisälle.

3.2 Datan esikäsittely

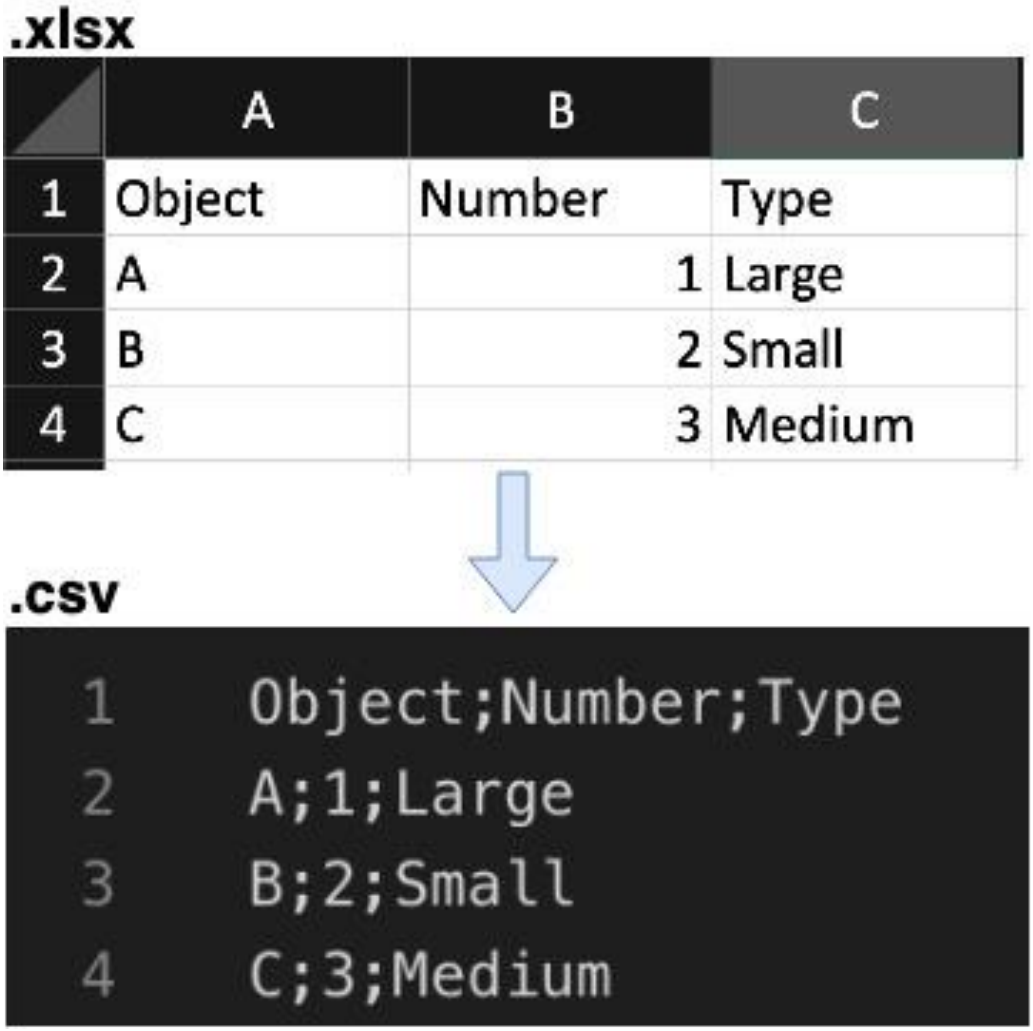
Asiakas tarjosi tarvittavan datan .xlsx tiedostomuodossa, jonka pystyi avaamaan, esimerkiksi Microsoft Excel -sovelluksessa. Asiakkaan tarjoama data sisälsi tietoa kahdelta vuodelta, ja rivejä datassa oli yli satatuhatta kappaletta. Jokainen rivi sisälsi yhden työntekijän tehtävän ja tehtävään kuluneen ajan.

Ensimmäinen askel datan esikäsittelyyn oli tunnistaa olennaiset tekijät, jotka vaikuttivat työntekijöiden tehtävän kestoon. Olennaiset tekijät on havainnollistettu taulukossa 2.

Taulukko 2. Datan olennaiset tekijät

Tekijä **(* käytetään prototyypin toteutuksessa)**	**Merkitys neuroverkossa**
Työntekijän tunnus (*)	Pystytään huomiomaan työntekijät yksilöllisesti.
Tehtävän tunnus (*)	Pystytään tunnistamaan ajankesto tiettyyn tehtävään.
Tehtävän kesto (*)	Ulostulo arvo neuroverkolle.
Tehtävän aloitus- ja lopetusajankohta	Mahdollisuus selvittää ajankohdan vaikutus tehtävän valmistukseen (aamu, ilta, yö).
Päivämäärä	Mahdollisuus selvittää päivämäärän vaikutus tehtävän valmistukseen (pyhät, vuodenaika).

Seuraavaksi data muutettiin yksinkertaisempaan tiedostomuotoon, jotta sitä olisi helpompi käsitellä. Tiedosto muutettiin Micorosoft Excelissä .csv (comma-separated values) formaattiin. Havannoillistava kuva tiedostoformaatin muutoksesta on kuviossa 11.



Kuvio 11. Data CSV-muodossa

Tiedostomuodon muuttamisen jälkeen dataa tarkasteltiin yksityiskohtaisemmin, jotta voitaisiin löytää virhearvoja, poikkeavaisuuksia ja erilaisia ongelmakohtia. Tietoa tarkasteltiin sarake kerrallaan tutkimalla muun muassa sarakkeiden ääreisarvoja, tietotyyppejä ja uniikkeja arvoja, jotta olisi mahdollista tunnistaa ja suodattaa pois ei-haluttua tietoja. Havainnollistava kuva datan lukemisesta CSV-muodossa on kuviossa 12.

```
// Rewritten for visualization purpose
const fs = require('fs');

// Read data from .csv file
fs.readFile('data.csv', 'utf8', (error, data) => {
    if (!error) {
        // Defining column keys
        // First column contains value of id
        const ID = 0;

        // Splits data by new line
        const lines = data.toString().split('\n');

        // Loop each line in data
        lines.forEach(line => {
            // Splits line by semicolon
            // Loop each column in line
            line.split(';').forEach((value, column) => {
                switch (column) {
                    case ID:
                        // Do something specific to value of id
                        handleId(value);
                        break;
                }
            })
        })
    }
})
```

Kuvio 12. Datan lukeminen CSV-muodossa

Työntekijöiden tunnus -sarakkeen osalta tiedoissa ei ollut minkäänlaisia ongelmakohtia, joita olisi tarvinnut huomioida ennen datan varsinaista käsittelyä. Tehtävien tunnus -sarakkeessa jouduttiin suodattamaan kaikki rivit pois, missä ei ollut paikkansapitäviä tehtävätunnuksia esimerkiksi työntekijöiden merkkaamat vuoronaloitukset, jotka eivät sisältäneet tehtävän tunnuksia. Tehtävän kesto -sarakkeessa kuitenkin ilmeni suuria ongelmia tiedon laadun kanssa. Työntekijät olivat merkanneet tunteja eri tehtäviin vapaalla kädellä, joka aiheuttaa runsaasti poikkeavaisuuksia tehtävien kestossa. Erilaisia tapauksia ilmeni lukuisia, ja niitä oli mahdotonta selvittää ohjelmallisesti tai käsin, koska ei voida tietää miten työntekijä oli oikeasti suorittanut kyseisen tehtävän. Tyypillisempiä poikkeustilanteita on havainnollistettu taulukossa 3.

Taulukko 3. Tyypilliset poikkeustilanteet tehtävän kestossa

Poikkeustilanne	**Ongelma**
Työntekijä varaa useamman tehtävän kerrallaan ja suorittaa tehtävät peräjälkeen. Työntekijä merkkaa yhteen tehtävään kaikkien tehtävien keston.	Merkattu tehtävä saa muiden tehtävien keston summan ja merkkaamattomien tehtävien kestot eivät sisällä paikkansapitäviä arvoja.
Työntekijä merkkaa työpäivän aikana suoritettujen tehtävien keston vuoronaloitukseen.	Tehtävien kestot eivät sisällä paikkansapitäviä arvoja.
Variaatioita molemmista tilanteista	Tehtävien kestot eivät sisällä paikkansapitäviä arvoja.

Datan suodatuksen jälkeen kelvollisia rivejä jäi enää jäljelle muutamia kymmeniä eli huonolaatuista dataa oli runsaasti. Tässä vaiheessa voidaan olettaa, että haluttuun ongelmaan ei voida saada luotettavaa vastausta, koska datan määrä jäisi erittäin vähäiseksi suodatuksen jälkeen tai ilman järkevää suodattamista data olisi kelvotonta. Koneoppimisen hyödyntämiseen ja neuroverkon kouluttamiseen olisi tärkeää, että data olisi laadukasta ja sitä olisi paljon. Datan laadusta huolimatta opinnäytetyön prototyyppi toteutettiin, koska yhtenä tavoitteena oli tutustua koneoppimisen menetelmiin.

Seuraavaksi data käsiteltiin neuroverkkoa varten eli tavoitteena oli normalisoida data. Ennen normalisointia oli kuitenkin huomioitava, että tiedot olivat sarakekohtaisesti oikeassa tietotyypissä, esimerkiksi tehtävien kestot olivat käännetty merkkijonosta numeeriseen arvoon.

Normalisointiin koskien alla mainittuja tietoja ei ole käytetty prototyypin toteutuksessa vaan ne ovat uudelleen kirjoitettu kuvitteellisilla tiedoilla, jotta asiakkaan antamia tietoja ei konkreettisesti tuoda esille opinnäytetyön raportissa. Tavoitteena on kuitenkin esittää datan käsittelyprosessia ennen kuin data syötetään neuroverkolle.

Kuviossa 13 esitetään luodun tietojoukon rakenne. Tietojoukko koostuu objekteista, jotka sisältävät kolme ominaisuutta, väri (color), äänien määrä (votes) ja suosittu status (popular).

```
// Mock data for neural network
const data = [
    { color: 'blue', votes: 10, popular: false },
    { color: 'yellow', votes: 50, popular: false },
    { color: 'red', votes: 170, popular: true },
];
```

Kuvio 13. Datan esittely normalisointia varten

Neuroverkon toimiakseen, sille täytyy syöttää koulutussarja, joka koostuu sisääntuloista ja ulostuloista. Tässä tapauksessa sisääntuloja ovat tietojoukon väri ja äänien määrä, ja ulostulo on suosittu status. Havainnollistava kuva tuloista on kuviossa 14.

```
// Training set for neural network before normalization
const trainingSet = [
    {
        input: ['blue', 10],
        output: [false]
    },
    {
        input: ['yellow', 50],
        output: [false]
    },
    {
        input: ['red', 170],
        output: [true]
    }
];
```

Kuvio 14. Koulutussarja ennen normalisointia

Neuroverkko ei kuitenkaan ymmärrä mitä tietojoukon arvot konkreettisesti ovat, vaan se ymmärtää vain yhden tulon, joka sisältää arvoja 0 ja 1 väliltä. Lisäksi tulojen on oltava kiinteän kokoisia, jotta kaikki tulot olisivat saman pituisia.

Väri voidaan normalisoida kategorioihin. Kuviossa 14 esitetty tietojoukko sisältää kolme uniikkia väriä ja jokainen väreistä voidaan ilmaista binäärimuuttujana. Värit voidaan esittää käyttämällä kolmea bittiä. Normalisointi kategorioihin on havainnollistettu taulukossa 4.

Taulukko 4. Normalisointi kategorioihin

Alkuperäinen	**Normalisoitu**
”blue”	1, 0, 0
”yellow”	0, 1, 0
”red”	0, 0, 1

Äänien määrä voidaan normalisoida lukujen 0 ja 1 välille hyödyntämällä yhtälöä 1. Jos luvun minimi ja maksimi arvo on tiedossa, niitä voidaan käyttää tietojoukon pienimmän ja suurimman arvon sijaan.

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (1)$$

missä z_i = normalisoitu arvo

x_i = alkuperäinen arvo

$\min(x)$ = pienin x:n arvo

$\max(x)$ = suurin x:n arvo

Taulukossa 5 on havainnollistettu äänien normalisointia ja lisäksi se sisältää kuviossa 14 esitetyt äänien arvot.

Taulukko 5. Numeerisen arvon normalisointi

10, 50, 170 **min(x) = 10** **max(x) = 170**	
Alkuperäinen (x_i)	**Normalisoitu (z_i)**
10	$\frac{10-10}{170-10} = \mathbf{0}$
50	$\frac{50-10}{170-10} = \mathbf{0.25}$
170	$\frac{170-10}{170-10} = \mathbf{1}$

Suosittu statuksen normalisointi tapahtuu yksinkertaisemmin. Statusta voidaan esittää bittinä. Havainnollistettu taulukossa 6.

Taulukko 6. Totuusarvon normalisointi

Alkuperäinen	**Normalisoitu**
true	1
false	0

Normalisoinnin jälkeen koulutussarja on käsitelty neuroverkolle ymmärrettävään muotoon. Koulutussarjan sisään- ja ulostulo normalisoinnin jälkeen on havainnollistettu kuviossa 15.

```
// Mock data for neural network
const trainingSet = [
    {
        input: ['blue', 10],
        output: [false]
    },
    {
        input: ['yellow', 50],
        output: [false]
    },
    {
        input: ['red', 170],
        output: [true]
    }
];
```

Kuvio 15. Koulutussarja normalisoinnin jälkeen

3.3 Neuroverkon koulutus

Datan normalisoinnin jälkeen data jaettiin osiksi: koulutussarjaan, joka sisälsi noin 80% datasta, ja testisarjaan, joka sisälsi noin 20% datasta. Neuroverkkoa koulutettiin koulutussarjalla ja neuroverkon toiminnallisuutta testattiin testisarjalla. Testisarja antoi puolueettoman arvioinnin koulutetun neuroverkon sovituksesta, koska testisarja ei sisältänyt samoja tietoja kuin koulutussarja.

Neuroverkon tyypiksi valittiin feedforward-neuroverkko, joka sisältää backpropagation toiminnon. BrainJs sisältää myös version feedforward-

neuroverkosta, joka käyttää GPU:ta CPU:n sijasta. Feedforward-neuroverkko pystyy tunnistamaan yksinkertaisia asioita hyvin, mutta sillä ei ole muistia edellisistä toiminnoistaan.

Feedforward-neuroverkolle määritettiin asetukset, jossa sille pystyttiin määrittämään aktivointifunktio, ja piilotettujen kerroksien ja solmujen määrä. Havainnollistava kuva neuroverkon alustuksesta ja sen asetuksista kuviossa 16.

```
// Neural network initialized and its config
const net = new brain.NeuralNetwork({
    activation: 'sigmoid', // Activation function
    hiddenLayers: [4], // 2 Hidden layers, both layer has 4 nodes
});
```

Kuvio 16. Neuroverkon alustus ja sen asetukset

Toteutuksessa asetuksiin määritettiin sigmoid-aktivointifunktio, joka on myös oletuksena päällä. Sigmoid-funktio esiintyy luvun 0 ja 1 välillä, siksi sitä käytetään erityisesti malleissa, joissa on ennustettava todennäköisyyttä ulostulona. Piilotettuja kerroksia asetettiin ainoastaan yksi, mutta kerroksen solmujen määrää säädettiin toteutuksessa vaihtelevasti. Oletuksena piilotettuja kerroksia on yksi ja kerroksen solmuja on sisääntulon pituuden verran.

Neuroverkon alustuksen jälkeen neuroverkkoa ruvettiin kouluttamaan sen koulutusfunktion avulla. Koulutusfunktion ensimmäisen argumentti sisältää koulutussarjan. Kun koulutetaan feedforward-neuroverkkoa, koulutussarjan jokaisessa kuviossa tulisi olla sisääntulo ja ulostulo (ulostulo). Molemmat tuloista voivat sisältää joukon numeroita taulukossa tai yksittäisen arvon lukujen 0 ja 1 välillä. Toisena argumenttina koulutusfunktio ottaa vastaan koulutusasetukset. Koulutusfunktiota on havainnollistettu kuviossa 17.

```
// Neural network training function
net.train(
    // Training set
    [
        {
            input: [1, 0, 0, 0],
            output: [0]
        },
        {
            input: [0, 1, 0, 0.25],
            output: [0]
        },
        {
            input: [0, 0, 1, 1],
            output: [1]
        }
    ],
    // Training options
    {
        log: true,  // Logs training error for each iteration
        logPeriod: 10, // Iterations between logging
        iterations: 100, // The maximum times to iterate the training set
        errorThresh: 0.005, // The axxeptable error percentage from training set
        learningRate: 0.3, // Scales with delta to effect training rate
        callback: null, // A periodic call back that can be triggered while training
        callbackPeriod, // Iterations between callbacks
        timeout: Infinity // Milliseconds to train for
    }
);
```

Kuvio 17. Neuroverkon koulutusfunktio

Neuroverkko lopettaa kouluttamisen, kun yksi seuraavista kriteereistä täyttyy: koulutusvirhe on ylittänyt virheen alarajan (errorThresh), iterointien enimmäismäärä on saavutettu (iterations) tai jos aikaraja ylittyy (timeout). Oletuksena koulutusfunktio ei ilmoita mitään koulutusprosessista ennen kuin toiminto on loppunut. Kun lokiarvo (log) asetetaan todeksi, koulutusfunktio tulostaa sen hetkisen koulutusvirheen. Koulutusvirheen pitäisi laskea koulutuksen edetessä. Lokien tulostus tiheyttä voidaan säätää määrittämällä lokin jakson pituus (logPeriod). Koulutustaso (learningRate) vaikuttaa siihen, kuinka nopeasti neuroverkko kouluttautuu. Koulutustason arvoksi voidaan asettaa luku 0 ja 1 väliltä. Liian suuren koulutustason määrittäminen saattaa aiheuttaa neuroverkossa helposti ylisovitusta. Oletuksena koulutustaso on arvoltaan 0.3.

Toteutuksessa kokeiltiin muuttaa koulutustasoa, virheen alarajaa, iteraatioiden määrää, ja piilotettujen kerroksien ja solmujen määrää, mutta merkittäviä eroavaisuuksia lopputuloksessa ei ilmennyt johtuen huonolaatuisesta datasta. Positiivisena huomiona kuitenkin koulutusvirhe laski jokaisen iteraation aikana, ja koulutusvirhe ei ollut liian suuri iteraatioiden jälkeen. Jos virhearvo jää suureksi, se yleensä on merkki siitä, että neuroverkko ei saa selkoa syötetystä datasta. Havainnollistava kuva neuroverkon koulutusvirheen tulostuksesta on kuviossa 18.

```
iterations: 89, training error: 0.021275772017071 88
iterations: 90, training error: 0.02126560582878427
iterations: 91, training error: 0.021255644665938007
iterations: 92, training error: 0.02124588843230728
iterations: 93, training error: 0.021236334341629807
iterations: 94, training error: 0.02122697949123659
iterations: 95, training error: 0.021217819599301926
```

Kuvio 18. Neuroverkon koulutusvirheen tulostus

Kouluttamisprosessin jälkeen neuroverkolle ajettiin testisarja, josta saatiin ennustettua ulostulon arvo. Alkuperäinen ja ennustettu arvo säilytettiin, jotta lopputulosta voitaisiin vertailla alkuperäisen ja ennustetun arvon välillä. Testisarjan ajoa on havainnollistettu kuviossa 19.

```
// Looping test set for trained neural network
testSet.forEach(obj => {
    // Each input in test set is run in trained neural network
    const output = net.run(obj.input);

    // Original and predicted output pushed to array for comparison
    testSetResults.push({
        original: obj.output,
        result: output
    })
});
```

Kuvio 19. Testisarjan ajo

4 Tulokset ja kokemukset

4.1 Tulokset

Lopputuloksena työntekijän tehtävän kestoa ei onnistuttu ennustamaan tarpeeksi tarkasti, jotta sitä voitaisiin hyödyntää käytännössä luotettavasti. Tulokset eivät vastanneet toisiaan alkuperäisen arvon ja neuroverkon ulostulon kanssa. Huonolaatuisella datalla oli suuri merkitys neuroverkon koulutusprosessissa, joten tuloksien arvot olivat erittäin ailahtelevia. Syynä datan laaduttomuuteen oli tietojen syöttövaiheessa, jossa työntekijät pystyivät merkkaamaan tehtävien keston liian vapaamuotoisesti. Seurauksena data sisälsi erittäin paljon hajontaa tehtävien keston välillä, jossa työntekijöiden tunnukset ja tehtävät olivat samoja.

Toteutus ei kuitenkaan ollut turha, vaikka prototyyppiä ei saatu toimimaan halutulla tavalla. Lopputuloksena kuitenkin onnistuttiin selvittämään asiakkaalle ongelmakohtia työntekijöiden tiedonsyötössä. Jos kerättyä tietoa haluttaisiin hyödyntää tekoälyratkaisussa, missä tiedon laatu ja runsaus on tärkeää, olisi syytä validoida, ehkäistä ja ohjeistaa työntekijöiden tietojensyöttöprosessia.

4.2 Kokemukset

Prototyypin toteuttaminen oli mielekästä, koska aikaisempi kokemus aiheeseen oli vain pintaraapaisu mitä prototyypissä toteutettiin. Lisäksi prototyyppi pystyttiin toteuttamaan vastamaan reaalimaailman esimerkkiä, missä tieto oli kerätty teollisuusyrityksen tietokannasta. Myös kiinnostus tehokkuuden parantamiseen ja teollisuuden digitalisoimiseen teki toteutuksesta erittäin mielenkiintoista.

Teknologian valinta toteutukseen soveltui hyvin käyttötarkoitukseen, mutta tulevaisuudessa olisi hyödyllisempää käyttää alustaa, joka menisi vielä syvemmälle teoriaan ja eri toiminnollisuuksiin. Kattavampi teknologia olisi Google Brain tiimin kehittämä Tensorflow alusta.

Datan analysointi ja neuroverkon koulutusprosessi on antanut paljon uusia näkökulmia, kuinka tietoa voitaisiin hyödyntää monella eri tapaa. Tietoa aiheesta oli runsaasti ja monipuolisesti, ja varsinkin suurella yhteisöllä oli positiivinen vaikutus erilaisten ratkaisujen etsinnässä, koska monet muutkin olivat törmänneet samanlaisiin kysymyksiin.

5 Pohdinta ja jatkotoimenpiteet

5.1 Pohdinta

Tekoälyratkaisut ja koneoppiminen ovat puhuttaneet paljon esimerkiksi sosiaalisessa mediassa ja erilaisissa asiantuntijafoorumeissa. Tekoäly on ottanut suuren roolin markkinoilla ja teollisuudessa, jonka seurauksena tekoälyratkaisut tulevat yleistymään runsaasti lähitulevaisuudessa.

Tekoäly pyrkii jäljittelemään vaativia toimintoja, jotka ovat usein tyypillisesti ihmisille. Suurena etuna kuitenkin on, että toiminnot pystytään automatisoimaan, joten erilaisilla ratkaisulla pystytään parantamaan yritysten liiketoimintaa ja yksilöiden arkea.

Koneoppimisen menetelmillä pystytään hyödyntämään dataa tehokkaasti, joka olisi ihmisille mahdotonta saavuttaa, koska kerätty tieto on tyypillisesti jatkuvasti kasvavaa, vaikealukuista ja ennen kaikkea tietoa on paljon. Kerättyä tietoa voidaan syöttää neuroverkoille, jonka pohjalta niitä pystytään kouluttamaan ja saamaan lopputuloksena haluttuun ongelman vastauksia. Kuitenkin erilaisilla neuroverkoilla ja koulutusprosessin asetuksilla on suuri merkitys lopputulokseen. Eli on tärkeä valita oikeanlaiset menetelmät tietyn tyyppisen ongelman ratkaisemiseksi. Laadukas data ja hyvin koulutettu neuroverkko mahdollistaa, että kerätystä tiedoista pystytään selvittämään uutta tietoa, jota voidaan hyödyntää tarkoituksen mukaan.

Tekoäly on kuitenkin herättänyt paljon keskustelua myös sen vaaroista. Monet sosiaalisen median alustat toimivat itsenäisesti erilaisten algoritmien avulla kohdemarkkinoinnissa. Kohdistetun tiedon avulla pystytään harjoittamaan esimerkiksi sosiaalista manipulointia levittämällä propagandaa yksilöllisesti. Lisäksi eri alustat keräävät huomattavasti yksilöityä tietoa käyttäjän toiminnoista, joten henkilön yksityisyyskin on vaarassa. On tärkeää ymmärtää, että tehokkailla teknologioilla on vaaransa varsinkin silloin kun sillä on vaikutusta yhteiskuntaan ja suureen massaan.

5.2 Jatkotoimenpiteet

Prototyyppiä olisi voinut jatkokehittää paljon pidemmälle, jos data olisi ollut laadukkaampaa. Toteutusta olisi voinut laajentaa, muun muassa seuraamalla työntekijöiden tehtävien päivän ajankohtaa. Huomioimalla onko tehtävä suoritettu aamu-, ilta- vai yövuorossa, jonka pohjalta olisi voinut analysoida onko päivän ajankohdalla positiivista vai negatiivista vaikutusta työntekijän tehtävän suorittamaan. Lisäksi olisi voinut myös analysoida viikonpäivien, pyhien ja vuodenaikojen vaikutusta tehtävien suorittamiseen.

Lopputuloksia olisi voinut myös visualisoida työntekijä- ja tehtäväkohtaisesti. Kuitenkin visualisoinnissa pitäisi huomioida, että tieto sisältää runsaasti tietoa asiakkaan yrityksestä ja sen työntekijöistä. Joten tiedon hyödyntäminen ja esittäminen julkisesti olisi väärin.

Lähteet

Almaliki, Z. 2019. Standardization vs Normalization. Mediumin verkkosivu. Viitattu 9.4.2019. https://medium.com/@zaidalissa/standardization-vs-normalization-da7a3a308c64

Artificial Intelligence vs. Machine Learning vs. Deep Learning. N.d. Skymind verkkosivut. Viitattu 1.3.2019. https://skymind.ai/wiki/ai-vs-machine-learning-vs-deep-learning

Artifiial Intelligence – What it is and why it matters. N.d. SAS Analytics Insights. Viitattu 19.2.2019. https://www.sas.com/en_us/insights/analytics/what-is-artificial-intelligence.html.

Collapick. 2019. Collapick Company Oy verkkosivut. Viitattu 1.3.2019. https://www.collapick.com/fi-new/etusivu

Data Mining. N.d. Techopedian verkkosivu. Viitattu 3.3.2019. https://www.techopedia.com/definition/1181/data-mining

Data Mining vs. Machine Learning: What's The Difference? 2017. Import.io. Viitattu 18.2.2019. https://www.import.io/post/data-mining-machine-learning-difference.

Data Mining - What it is and why it matters. N.d. SAS Analytics Insigths. Viitattu 3.3.2019. https://www.sas.com/en_us/insights/analytics/data-mining.html

Katsaus neuroverkkomallintamiseen. N.d. Aalto-yliopiston verkko-opinnot. Viitattu 8.3.2019. http://salserver.org.aalto.fi/vanhat_sivut/Opinnot/Mat-2.3130/verkkoekskut/verkkoxq05.html

Lohr, S. 2014. For Big-Data Scientists, "Janitor Work" Is Key Hurdle to Insights. The New York Times. Viitattu 18.2.2019. https://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html

Marr, B. 2016. The Top 10 AI and Machine Learning Use Cases Everyone Should Know About. Forbesin verkkosivu. Viitattu 9.4.2019. https://www.forbes.com/sites/bernardmarr/2016/09/30/what-are-the-top-10-use-cases-for-machine-learning-and-ai/#77a9796694c9

Machine Learning and Data Quality. N.d. Spotlessdata verkkosivun blogiteksti. Viitattu 8.4.2019. https://spotlessdata.com/blog/machine-learning-and-data-quality

Machine Learning – What it is and why it matters. N.d. SAS Analytics Insights. Viitattu 18.2.2019. https://www.sas.com/en_ae/insights/analytics/machine-learning.html.

Mayo, H., Punchihewa, H., Emile, J. & Morrison, J. 2018. History of Machine Learning. Viitattu 18.2.2019. https://www.doc.ic.ac.uk/~jce317/history-machine-learning.html#top.

Mehta, A. 2019. A Complete Guide to Types of Neural Networks. Viitattu 8.4.2019. https://www.digitalvidya.com/blog/types-of-neural-networks/

Ray, S. 2018. History of AI. Viitattu 1.3.2019. https://towardsdatascience.com/history-of-ai-484a86fc16ef

Trask, M. 2018. General vs Narrow AI. Hackernoon. Viitattu 19.2.2019. https://hackernoon.com/general-vs-narrow-ai-3d0d02ef3e28.

Turner, V. 2014. The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things. Executive Summary. Viitattu 18.2.2019. https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm

What is big data? N.d. Oracle Big data. Viitattu 18.2.2019. https://www.oracle.com/big-data/guide/what-is-big-data.html.

What is Deep Learning? N.d. Mathworksin artikkeli. Viitattu 6.3.2019. https://www.mathworks.com/discovery/deep-learning.html

What is Machine Learning? A definition. N.d. Expret System blog. Viitattu 1.3.2019. https://www.expertsystem.com/machine-learning-definition/

What Is a Neural Network? N.d. Mathworksin artikkeli. Viitattu 5.3.2019. https://www.mathworks.com/discovery/neural-network.html