Neuroverkon käyttö urheilutulosten ennustamisessa

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Ylemmän ammattikorkeakoulututkinnon opinnäytetyö

HAMK Visamäki, Älykkäät palvelut

Kevät, 2019

Harri Maunu

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Älykkäät palvelut Tiivistelmä

HAMK Visamäki

Tekijä Harri Maunu Vuosi 2019

Työn nimi Neuroverkon käyttö urheilutulosten ennustamisessa

Ohjaajat Vesa Salminen

TIIVISTELMÄ

Tämän opinnäytetyön tarkoituksena on selvittää koneoppimisen ja etenkin neuroverkon soveltuvuutta urheilutulosten ennustamisessa. Neuroverkot haastavat urheilun asiantuntijoita enenemissä määrin niiden rajattoman kapasiteetin ja objektiivisen lähestymistapansa ansiosta. Opinnäytetyössä käsitellään myös sijoitusjärjestelmiä ja niitä käytetään myös datan lähteenä neuroverkon käytännön toteutuksessa.

Keskeisenä tavoitteena on selvittää, pystyykö yksinkertainen neuroverkko valitsemaan oikean painotuksen joukkueen yleisen tason ja viimeaikaisten otteluiden, eli niin sanotun kuntopuntarin välillä ja ennustamaan amerikkalaisen jalkapallon otteluiden voittajat NFL:ssä.

Neuroverkolle annettiin datalähteeksi NFL:n kaikki sarjaottelut vuodesta 1968 lähtien. 12 934 amerikkalaisen jalkapallon ottelun perusteella, neuroverkon oppima malli onnistui ennustamaan oikein 64,45 % tarkkuudella kauden 2018–2019 otteluiden voittajat.

Tuloksista voidaan todeta, että neuroverkko soveltuu erinomaisesti ottelutulosten ennustamiseen. Johtuen vedonlyöntiyhtiöiden asettamasta marginaalista voitolliseen vedonlyöntiin vaaditaan kuitenkin vielä parempia ennustuksia. Lisäämällä datalähteisiin enemmän muuttujia, voitaisiin mahdollisesti opinnäytetyössä rakennetun neuroverkon kaltaista koneoppimismallia kehittää.

Avainsanat Koneoppiminen, neuroverkko, Elo – luku, urheilutulosten ennustaminen

Sivut 40 sivua ja liitteitä 0 sivua

HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Name of Degree Programme

HAMK Visamäki

Abstract

Author Harri Maunu

Year 2019

Subject Forecasting the outcomes of sports events with neural networks.

Supervisors Vesa Salminen

ABSTRACT

The purpose of this thesis is to find out the suitability of machine learning and especially the neural network in predicting sports results. Neural networks are challenging sports experts because of their limitless capacity and objective approach. The thesis also deals with ranking systems, and they are also used as a data source in the practical implementation of the neural network.

The key goal is to find out if a simple neural network can choose the right weight between the team's overall level and recent matches, the so-called fitness level, and forecast the winners of American football matches in the NFL.

As a data source on neural network there was all NFL matches since 1968. Based on 12,934 American football matches, the model that neural network learned, it was able to correctly predict the winners of season 2018–2019 matches with 64,45% accuracy.

The results show that the neural network is excellent for predicting match results. However, due to the margin set by betting companies, even better predictions are required for profitable sports betting. By adding more variables to the data sources, a machine learning model used on a thesis, neural network could possibly be developed.

Keywords Machine learning, neural network, Elo rating, forecasting the outcomes of sports events

Pages 40 pages and appendices 0 pages

Sisällys

Kuvat, taulukot ja kaavat

1 Johdanto

Muutaman vuosikymmenen kestänyt digiloikka alkaa saavuttaa pisteen, jossa alkaa syntyä uusia hallitsemattomia ongelmia. Palveluiden siirtyminen digitaaliseksi loi tilanteen, josta alkoi väistämätön hukkuminen tiedon valtamereen - ja tuo talteen otetun tiedon määrä jatkaa kasvamista kiihtyvällä tahdilla. Oli kyse lääketieteen kuvantamisista tai liikennekameroiden tallennuksista, ne ovat meille yhteiskuntana arvokasta dataa, jos vain ehtisimme ottaa niistä kaiken hyödyn irti. Emme vain valitettavasti ehdi. Tiedon käsittely on nopeampaa päätelaitteella kuin ruutuviholla, mutta ei silti riittävän nopeaa. Koneoppiminen saattaa olla ongelmiimme ratkaisu.

Digitalisaatio poisti monta työväenluokan työtä, mutta nyt on koneoppimisen vuoro härnätä keskiluokan ammatteja. Jos koskaan työsi on tuntunut tylsältä, se luultavasti johtuu siitä, että siinä on liikaa toistoa. Koneoppiminen rakastaa toistoa ja on valmis viemään työpaikkasi. (Wiele, 2019)

Myynti odotti ennusteet ja nyt tuijottelet Excel-taulukkoa ja yrität etsiä sieltä syytä, miksi näin tapahtui tai junavuoroja olikin tarpeeseen nähden liian vähän ja nyt seilailet sosiaalista mediaa etsien sieltä keskusteluja aiheesta - mihin ihmiset olivat menossa tai mistä he olivat tulossa. Tällaisissa toiminnoissa koneoppiminen on parhaimmillaan, se etsii poikkeavuuksia, samankaltaisuuksia tai luo ennustuksia, niitä samoja, joita ihminenkin voisi tehdä, ainoastaan huomattavasti nopeammin ja tehokkaammin.

Koneen tekemät päätökset ovat valistuneempia, mutta myös objektiivisempia ja tästä syystä koneoppiminen soveltuu erinomaisesti urheilutulosten ennustamiseen. Koneella kun ei ole suosikkijoukkuetta tai se ei sääli altavastaajaa.

Urheilussa etenkin suurseurat ovat jo siirtyneet dataohjattuun valmentamiseen ja lajit kehittyvät tätä myötä nopealla tahdilla. Pelaajia valmennetaan kentällä liikkumaan ainoastaan alueilla, joissa heillä on tilastollisesti kannattavinta liikkua ja pelipäivän kokoonpanoon voidaan tehdä oleellisia muutoksia edellisyön unen laadun perusteella.

Koska nykytekniikalla on jo mahdollista luoda data-analyysejä videoista, tämä mahdollistaa sen, että pelaajien suorituskyvystä kertova data ei ole ainoastaan joukkueen sisäisessä käytössä. Myös ammattimainen vedonlyönti kehittyy nopeasti ja saatavilla olevan datan käsittely on siirtymässä ihmisten ammattitaidosta ja intuitiosta, koneiden laskettavaksi. (European Business Review, 2021)

Tässä opinnäytetyössä pyrimme selvittämään yksinkertaisen neuroverkon mahdollisuuksia ennustaa otteluiden voittajia. Kohteena käytämme amerikkalaisen jalkapallon ylintä sarjaa NFL:ää ja pyrimme tunnetun luokittelujärjestelmän (puhek. Ranking-järjestelmä) avulla luomaan dataa neuroverkolle. Koska kyseessä on verrattain loukkaantumisherkkä laji ja lepopäivien vaikutuksesta lopputulokseen on merkittäviä tutkimuksia, käytämme datalähteenä myös lepopäivätietoja. (Fantasy Labs, 2015). Tavoitteena on selvittää, voiko neuroverkko luoda toimivan mallin, joka rakentuu ainoastaan tulostietojen pohjalta luodun ranking-järjestelmän ja lepopäivätietojen varaan.

2 Koneoppiminen

Koneoppiminen itsessään on laaja käsite ja se voidaan määritellä monella eri tapaa. Pohjimmiltaan koneoppiminen on tilastotiedettä. Koneoppiminen käsittää erilaisia menetelmiä, joilla datasta voidaan etsiä tietoa, johdonmukaisuuksia tai poikkeavuuksia. (Elements of AI, n.d.)

Koneoppimisen ja tietokonesovelluksen ero on häilyvä. Koneoppimista kutsutaankin usein sovellus 2.0:ksi (engl. software 2.0), sillä se on ikään kuin kehittyneempi versio perinteisestä sovelluksesta. Perinteinen tietokoneella tehty sovellus tai skripti voi toimia ilman koneoppimista, mutta ei koskaan toisinpäin. Esimerkiksi käsin kirjoitetun tekstin tunnistaminen on mahdollista toteuttaa perinteisenä sovelluksena, sekä myös koneoppimisen menetelmin. Tällaiset tapaukset ovat sellaisia tilanteita, joissa perinteinen tietokonesovellus on erittäin työläs toteuttaa, kun taas koneoppimisella ongelma on ratkaistavissa huomattavasti pienemmällä työllä.

Perinteinen tietokonesovellus on ihmisen tekemä joukko määritelmiä, “jos x, niin y”. Koneoppimisessa on kyse siitä, että kone etsii joko syitä tai seurauksia joita ihminen ei itse

löytäisi, tai niitä olisi äärimmäisen hidasta löytää. Eli voisikin kärjistää asian niin, että tietokonesovelluksessa ohjelmoija osoittaa tietokoneelle syy-seuraus-suhteessa olevat x:n ja y:n, kun taas koneoppimissa kone etsii datasta x:n ja y:n ja osoittaa ihmiselle näiden korrelaation. (Singhal, 2019)

Tietokonekäsittelyopin tutkija Andrej Karpathyn mukaan ollaan menossa siihen, ettei koodausta enää tarvita ja koodareista on tulossa tietynlaisia “koodin mahdollistajia” tai “koodin ohjaajia” syöttäessään tietoa valmiiksi tuotetuille algoritmeille. (Huston, 2019)

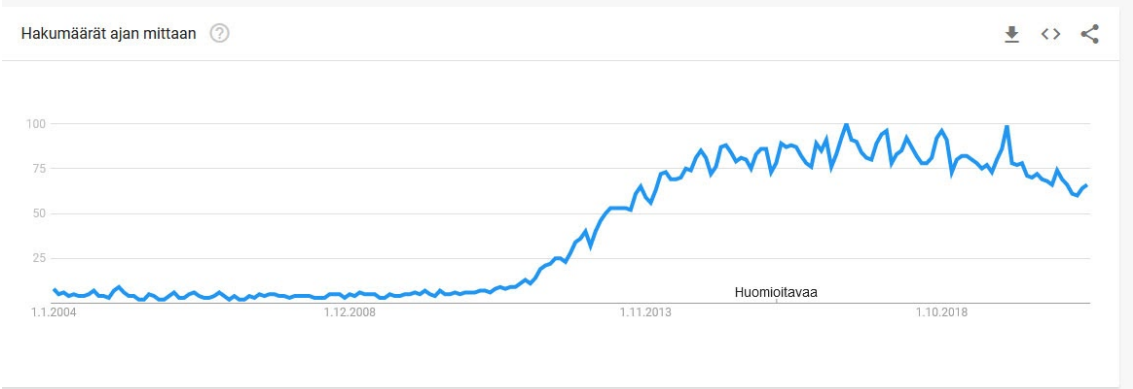
Ohjelmoimalla tehdään koodiryhmiä, algoritmeja. Algoritmit käsittelevät suuria datamääriä, big dataa. Data-analytiikka käsittelee lopputuotoksia ihmisen ymmärrettävään tai hyödynnettävään muotoon ja koko tätä prosessia, sen valvomista ja suunnittelua kutsutaan koneoppimiseksi.

Tässä luvussa käydään läpi koneoppimisen pääelementit.

2.1 Big data

Datan käsittely tietotekniikan avulla ei sinällään ole uusi keksintö, ei ainakaan sen uudempi kuin tietokoneet ylipäänsä, mutta kuten big data jo sanana viittaa, tällä tarkoitetaan huomattavan suurta määrää dataa.

Kuva 1. "Big data" - hakujen määrä Googlessa vuosina 2004–2020.



Suurien tietomassojen käsittely on noussut uudeksi tietotekniikan trendiksi. Kuten voidaan nähdä, “big data”-hakujen määrä on kasvanut Googlessa merkittävästi vuosien 2011–2012 vaihteessa (Kuva 1). Pilvipalvelut ja suuret tallennuskapasiteetit suhteellisen alhaisin kustannuksin mahdollistavat suurien tietomäärien varastoinnin, käsittelyn ja analysoinnin. Vaikka Google hauissa big data löi läpi vasta vuonna 2011, big data – käsite esiteltiin jo vuonna 2005. (van Rijmenam, 2013).

Se, että mikä määrä dataa täyttää ”suuri” (engl. big) -sanan määritelmän on tietysti suhteellista ja siksi on jokseenkin mahdoton määritellä milloin big datan kausi alkoi vai onko se edes vielä alkanut. Vaikka mitään tarkkaa linjanvetoa ei voida tehdä, määrältään big datalla viitataan sellaiseen määrään dataa, joka olisi ihmiselle liian suuri määrä läpikäytäväksi, ainakin kohtuullisessa ajassa.

Oli big datan kausi sitten missä vaiheessa tahansa, voidaan olla yhtä mieltä siitä, että kaikkien saataville syntyy kovaa vauhtia uusia palveluja vanhojen tieltä, koska big data mahdollistaa sellaisia kehityksen suuntia, joista aikaisemmin ei olisi voinut puhuakaan. Siinä missä big datan myötä syntyy uusia innovaatioita ja liiketoimintamalleja, myös täysin uusia ammatteja. Myös datasta itsessään tulee kauppatavaraa. ”Tieto on valtaa” on vanha sanonta, mutta uusi sanonta voisikin olla ”tieto on rahaa”. (eWeek, 2019)

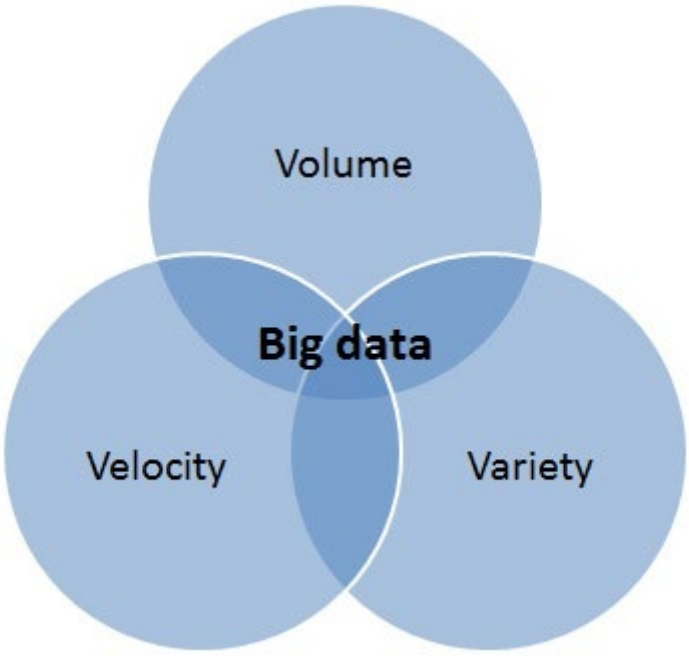
Jotta data olisi saatavilla suuremman joukon käyttöön, tarvitaan pilvipalvelua tiedon varastointiin. Pilvipalvelut ovat kaikessa yksinkertaisuudessa internetyhteyden yli käytettäviä tietokoneita. Suurten datamäärien käsittely vaatii myös laskentatehoa. Big datan käsittelyyn tarkoitetut pilvipalvelut tarjoavat loppukäyttäjälle sekä tallennustilaa, että tehokkaat pilvitietokoneet (engl. cloud computing).

Se minkälaisia palveluita data mahdollistaa, hyvänä esimerkkinä on sosiaalisen median palvelut. Palveluntarjoaja ei pyydä käyttäjältä maksua palvelun käytöstä, ja vastineeksi käyttäjät antavat valtavan määrän tietoa itsestä ja omasta käyttäytymisestä palveluntarjoajalle. Sosiaalisen median palvelut ovat siis suuren datamassan päällä toimivia ihmisten sosiaalisen vuorovaikutuksen mahdollistavia palveluita. Saamaansa dataa sosiaalinen media käyttää haluamallaan tavalla, kuten myymällä kohdennettua mainostilaa omassa palvelussaan. Suurimpien sosiaalisen median alustojen valtavan suuri valuaatio

selittyy siis niiden hallinnoiman datan ja sen kysynnän perusteella, eikä niinkään heidän kehittämän teknologian perusteella. (Salo, 2014. 6–9).

2.2 Datan ominaisuudet

Kuva 2. Kolme V:tä.



Big data voidaan jaotella ominaisuuksiltaan eri tavoin. Yleisimmin puhutaan kolmesta V:stä (Kuva 2): volyymi (engl. volume), vauhti (engl. velocity) ja vaihtelevuus (engl. variety), mutta myöhemmin on tullut mukaan myös muuntautuvuus (engl. Variability), luotettavuus (engl. veracity), visuaalisuus (engl. visualization) ja arvo (engl. value).

Volyymi (engl. volume) on käytettävissä oleva datan määrä. Datan määrä kasvaa eksponentiaalisesti joka vuosi kuvina, videoina, tietokantoina ja erilaisten sensoreiden keräämien tietojen muodossa. Lisääntyvä laitteiden kiinnittyminen tietoverkkoihin (IoT) lisää talteen otetun datan määrää. Datan määrää voidaan pitää koko big data -käsitteen pohjana. Vuonna 2020 arvioitiin syntyvän 59 zetatavua, eli 59 000 000 000 teratavua dataa. (IDC, 2020)

Vauhti (engl. velocity) määrittelee tiedon tuoreuden, eli sen kuinka nopeasti dataa syntyy ja kuinka nopeasti sitä prosessoidaan. Se kuinka reaaliaikaista data on, saattaa olla hyvinkin merkittävää, kun yrityksen toimintaa ohjataan datasta johdetun tiedon avulla. Esimerkiksi yritys saattaa muokata markkinointikampanjaa myyntilukujen perusteella käynnissä olevan jalkapallo-ottelun aikana. Tämäntyyppisissä tilanteissa dataa on pystyttävä analysoimaan välittömästi, eikä vasta tuntien kuluttua. (BBVA, 2020)

Monimuotoisuus (engl. variety) jakautuu jäsenneltyyn (engl. structured) tai jäsentelemättömään (engl. unstructured) dataan tai niiden sekoitukseen, osittain jäsenneltyyn dataan (engl. semistructured). Jäsennetty data on taulukoitua ja koneen luettavissa olevaa dataa, kun taas jäsentämätön on kuvia, videoita ym. ei koneen suoraa luettavissa olevaa dataa. Suurin osa olemassa olevasta datasta on jäsentelemätöntä. Selkein esimerkki osittain jäsennellystä datasta on älypuhelimella otettu valokuva. Kuvatiedosto itsessään on jäsentelemätöntä dataa, mutta koska tiedoston metatiedoissa on jäsenneltyä dataa, kuten tiedot kuvan ottohetkestä ja mahdollisesti GPS-sijaintitiedot, tämä tekee kuvatiedostosta osittain jäsenneltyä dataa. (ABCadda, 2020)

Näiden kolmen V:n lisäksi datan ominaisuuksia kuvailtu myös muilla tavoin, kuten muuntautuvuus, luotettavuus, visuaalisuus ja arvo.

Muuntautuvuus (engl. variability) sekoitetaan helposti vaihtelevuuteen (engl. variety), mutta kyse on hyvin eri asiasta. Kyse on yksittäisen ominaisuuden muuntautumisesta. Tilanteessa, jossa kerätty data on ominaisuuksiltaan koko tarkastelujakson sama, ympäröivät tekijät voivat kuitenkin vaikuttaa dataan niin, ettei se ole välttämättä enää vertailukelpoista. (YourTechDiet, n.d.)

Esimerkiksi tarkasteltaessa keihäänheiton ennätystuloksia, vuonna 1984 heitettiin 104,8 m, kun taas kaksi vuotta myöhemmin ennätysheitto oli lähes 20 metriä vähemmän. Kyseinen tapahtuma selittyy kilpailuissa käytettävän keihäsmallin muutoksella. (Track and Field Statistics, 2004)

Muuntautuvuuden kanssa hieman samankaltainen datan ominaisuus on luotettavuus (engl. veracity). Luotettavuuskysymys on erittäin tärkeä osa koneoppimista ja datan käsittelyä.

Suurista datamääristä voi olla hyvin vaikea todentaa datan todenmukaisuutta. Todenmukaisuus kysymyksen äärellä ollaan, kun tarkastellaan vaikka sähköpostilistalle ilmoittautuneita nimiä, kuten ”aku ankka” tai ”asdasd” tai silloin kuin vertaillaan lääketieteellisten kuvantamisten tuloksia huippuunsa varustellun länsimaisen sairaalan ja kehitysmaassa sijaitsevan pienen sairaalan välillä. (Impact, 2016)

Visuaalisuus (engl. visualization) on olennainen osa big datan käsittelyä, vaikka se ei koske varsinaista laskentatyötä. Tulee muistaa, että dataa käsitellään aina siksi, että on olemassa jokin ongelma, joka halutaan ratkaista ja näin ollen datan on oltava ihmisen ymmärrettävissä. Visuaaliset grafiikat, pistepilvet ja diagrammit ovat helpommin ymmärrettävissä kuin pelkät arvoja sisältävät taulukot.

Arvo (Value) on tärkein ja lopullinen osa koko big datan käsittelyä. Jotta datan käsittelystä on ollut jotain hyötyä, tulokset on annettava sellaisessa muodossa, että ne tuottavat arvoa ihmisille. Arvolla ei siis niinkään oteta kantaa itse dataan vaan siitä johdetun tiedon arvoon. (Impact, 2016)

2.3 Algoritmit

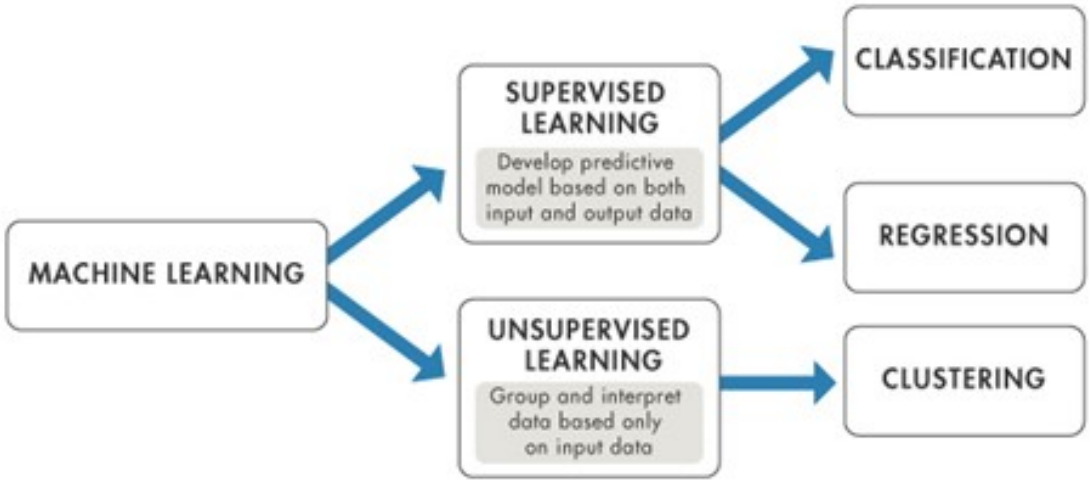
Algoritmit ovat kaikessa yksinkertaisuudessaan käskysarjoja. Peräkkäin, tai pikemminkin allekkain asetettuja käskyjä, joita tietokoneen tulee suorittaa. Tietokone ei luonnollisesti päättele tai osaa tehdä yksinkertaisiakaan oletuksia mistään vaan noudattaa orjallisesti sille annettuja käskyjä. Koska tietokoneiden elämään sisältyy paljon toistuvia ja toistettavia tapahtumia, näitä varten on luotu valtava määrä yleisesti tunnettuja käskysarjoja. Algoritmia muodostettaessa, eli ohjelmoitaessa, ei siis tarvitse keksiä aina polkupyörää uudelleen vaan usein voidaan käyttää, jo olemassa olevia käskysarjoja, osana kokonaisuutta. Tavoitteen saavuttamisen suhteen ei ole merkitystä mitä ohjelmointikieltä algoritmiin käytetään, jos algoritmi on virheetön, lopputulos on aina sama.

Tilannetta voisi verrata yleisesti tunnettuihin matemaattisiin kaavoihin. Pythagoraan lause antaa näppärästi oikopolun oikeaan vastaukseen kolmion sivun pituuksia laskettaessa, ilman, että aina tarvitsee todentaa syytä siihen, miksi kyseinen kaava toimii. Vaikka itse ohjelmaa

ajettaessa tietokone ei välitä kirjoititko kaiken itse vai käytitkö valmista käskysarjaa, sillä se joutuu tekemään joka tapauksessa saman määrän töitä, niin sillä on paljonkin merkitystä, kuinka paljon jouduit käyttämään aikaa algoritmin rakentamiseen. (Kokkarinen 2002, 15-16.)

Erilaisia algoritmeja käytetään laajasti etsittäessä johdonmukaisuuksia datajoukoista. Ei ole myöskään mikään harvinaisuus, että yhteen oppimisprosessiin käytetään monia erilaisia algoritmimalleja, kunnes parhaimpaan lopputulokseen päästään.

Kuva 3. Koneoppimismallien jakaantuminen eri käyttötarkoituksiin.



Koneoppimisessa tapa, jolla koneet ”oppivat”, jakavat algoritmit kahteen pääryhmään: Ohjattuun (engl. supervised) ja ohjaamattomaan oppimiseen (engl. unsupervised learning).

2.3.1 Ohjattu koneoppiminen (engl. supervised learning)

Ennen algoritmin ohjelmointia, tulee tunnistaa käytössä olevat tiedot, mitä niistä ollaan johtamassa, mikä on päämäärä ja mitä työkaluja tähän on järkevintä käyttää.

Ohjattu oppiminen tarkoittaa prosessia, jossa sekä lähtötilanne, että lopputulos ovat tiedossa ja vastaus löydetään vertaamalla ongelmaa, jo aikaisemmin tapahtuneeseen tilanteeseen.

Esimerkiksi internetin keskustelufoorumeilla on hyödynnetty tämän kaltaista koneoppimista. Laadukkailla keskustelufoorumeilla toimii ihmis-operaattoreita, jotka poistavat asiattomia kommentteja. Koneoppimisen avulla voidaan käydä läpi poistettuja viestejä, etsiä niistä yhtäläisyyksiä ja eroavaisuuksia asiallisiin viesteihin ja poistaa automaattisesti uudet viestit, jos ne sisältävät samankaltaista sisältöä kuin aikaisemmin poistetuissa viesteissä. Tällaisessa tilanteessa on kyse ohjatusta oppimista, koska lähtötilanne, eli viestin sisältö ja lopputulos, eli viestin poistaminen tai poistamatta jättäminen, ovat tiedossa.

Ohjattu oppiminen jaetaan kahteen kategoriaan regressioon (regression) ja luokitteluun (classification). Nämä kaksi kategoriaa kuuluvat ohjattuun oppimiseen, koska ne käyttäytyvät kausaalisesti vastaten suoraan kysymykseen suoralla vastauksella. Luokittelussa vastaus on mallia kissa, koira ja marsu tai vain yksinkertaisesti tosi (engl. true) tai epätosi (engl. false), kun taas regressiivisessa ongelmassa vastaus on reaali- tai liukuluku. Kuten esimerkiksi paino, hinta tai lämpötila. Oli vastaus luokittelun tai regression mukainen, yhteneväistä kuitenkin on se, että molemmissa muuttujalla on jokin nimike (engl. label) ja tälle ennalta annettu arvo (engl. value).

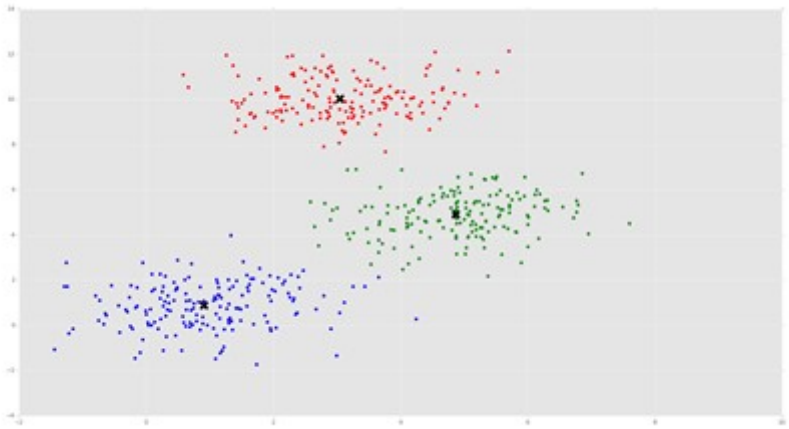
Luokitteluun (classification) tarkoitettuja algoritmimalleja, luokittimia (classifiers), on useita. Näistä tunnetuimpia ovat logistic regression, decision tree, support vector machines, neuroverkko (artificial neural networks) sekä Naive Bayes. (Wolff, 2020)

2.3.2 Ohjaamaton koneoppiminen (engl. unsupervised learning)

Ohjaamaton oppiminen (engl. unsupervised learning) on huomattavan paljon monimutkaisempi ja monitahoisempi sekä suunnitella että toteuttaa kuin ohjattu oppiminen. Ohjaamattomassa oppimisessa lähtödatassa ei ole vastauksia, joten siinä ei ole niinkään kysymys oikein vastausten löytämisestä, vaan oikeiden kysymysten.

Ohjaamaton oppiminen voidaan jakaa algoritmien perusteella kahteen eri ryhmään, klusterointiin (Clustering) ja assosiointiin (Association).

Kuva 4. Pistepilvi.



Klusterointi valitaan, kun kohdataan ongelma, jossa datasta pitää etsiä yhtäläisyyksiä. Klusteria voisi kutsua myös kertymäksi tai keskittymäksi, sillä kyseisessä menetelmässä etsitään moniulotteiseen koordinaatistoon syntyviä ryhmiä. Graafisesti esitettynä kyse on usein pistepilvistä, kuten kuvassa 4.

Erona ohjattuun oppimiseen, ohjaamattomassa oppimisessa data on jäsentelemättämässä muodossa. Ideana on siis mallintaa dataa tavalla, joka erottelee alkiot toisistaan ja löytää niistä yhtäläisyyksiä, sekä eroavaisuuksia. (IBM, n.d.)

2.3.3 Vahvistusoppiminen (engl. Reinforcement learning)

Vahvistusoppiminen oma mallinsa, vaikka se hieman muistuttaa ohjattua oppimista. Vahvistusoppimisessa on sekä sisään tuleva data, että tulos, kuten ohjatussa oppimisessa, mutta sillä hyvin merkittävällä eroavaisuudella, että tulos saadaan vasta jälkikäteen palautteen muodossa.

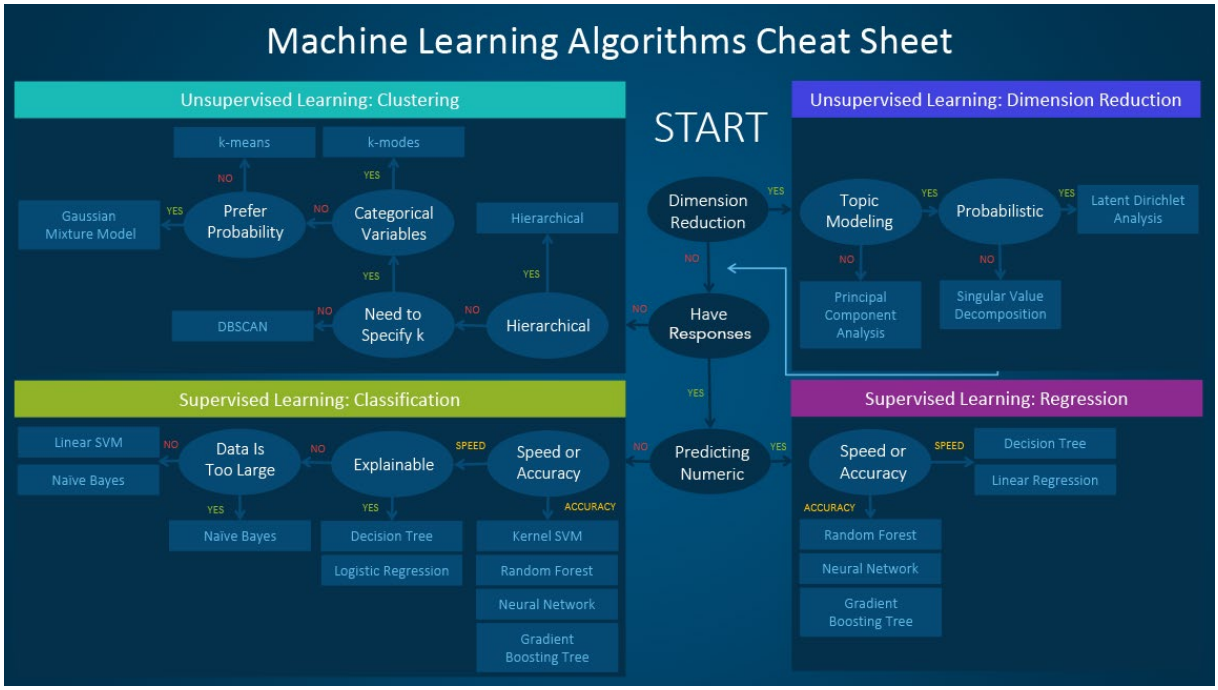
Vahvistusoppiminen saa siis nimensä siitä, että se pyytää ikään kuin vahvistuksen oman toiminnan oikeellisuudesta. Palaute on joko positiivinen, neutraali tai negatiivinen. Tällaista mallia käytetäänkin esimerkiksi tilanteessa, jossa rakennetaan tekoälyä dynaamiseen ympäristöön. Tällainen ympäristö voi olla vaikkapa liikenne ja koneoppimisen avulla opetetaan itsestään ajavaa autoa selviytymään liikenteessä. Loputon määrä erilaisia

muuttuvia tekijöitä tekisi käytännössä mahdotonta siitä, että autoon rakennettaisiin ainoastaan perinteisen sovelluksen tavoin ajotietokone tekemään valintoja. Liikennevalojen vaihtuminen ja kaistalla pysyminen on vielä melko yksinkertaisia asioita koodata sovellukseen, mutta muut tienkäyttäjät ja etenkin heidän tekevät ajovirheet ja niiden ennakointi tekevät tilanteesta monimutkaisempaa. (Guru99, n.d.)

2.3.4 Oikean algoritmin valitseminen

Koska koneoppimiseen on tarjolla laaja kirjo laskentatapoja, oikean valinnan tekeminen voi olla vaikeaa. Itse valinnan tekemisen voi ajatella jakautuvan kahteen vaiheeseen. Ensin on valittava ne tavat, jotka ylipäänsä ovat mahdollisia projektin suorittamiseen. Esimerkiksi ohjatun oppimisen algoritmeja ei voi käyttää ollenkaan, jos datassa ei ole tulos - arvoja. Kun käytettävissä olevat keinot on löydetty tai mahdottomat rajattu pois, voidaan valita tilanteeseen sopiva algoritmi. On yleistä, että algoritmeja käytetään useita ja valitaan laskennan jälkeen niistä parhaiten tehtävästä suoriutunut malli. (Li, 2020)

Kuva 5. Algoritmin valitseminen (lähde: sas.com).

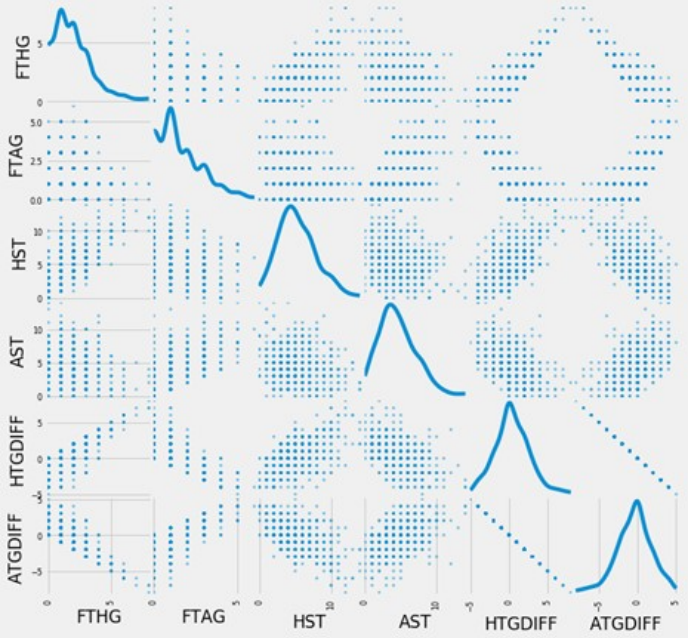


Myös itse datan määrällä, tai pikemmin sen muodolla on merkitystä valintaan. Lähtökohtaisesti parasta olisi, että dataa olisi käytössä riittävästi, mutta käytännössä tähän

ei kuitenkaan usein pääse vaikuttamaan, vaan on käytettävä sitä määrää dataa, joka on saatavilla, oli määrä mikä tahansa.

Itse datan tunteminen on tärkeää ja tätä voi edistää visualisointien avulla. Visualisoinnit voivat auttaa tunnistamaan datasta korrelaatioita. Parikoordinaatistosta (engl. pair plot) selkeiden korrelaatioiden löytäminen on melko yksinkertaista.

Kuva 6. Parikoordinaatisto.



Kuvassa 6 esimerkki datasta muodostetusta visualisoinnista. Kyseisessä kuvassa on jalkapallo-ottelun tilastoja. Attribuutit on jaettu X ja Y akselille ja jokainen yksittäinen ruutu sisältää oman pienemmän XY- koordinaatiston. Sekä X-, että Y-akselilta löytyvät samat attribuutit.

Kuvan attribuutit ovat lyhenteitä sanoista:

- FTGH (Full Time Home Goals) = Kotijoukkueen maalimäärä
- FTAG (Full Time Away Goals) = Vierasjoukkueen maalimäärä
- HST (Home Shots on Target) = Kotijoukkueen laukaukset kohti maalia
- AST (Away Shots on Target) = Vierasjoukkueen laukaukset kohti maalia
- HTGDIFF (Home Team Goal Difference) = Kotijoukkueen voittomarginaali
- ATGDIFF (Away Team Goal Difference = Vierasjoukkueen voittomarginaali

Tämäntyyppisessä datassa korrelaatiot saattavat olla itsestäänselvyys, jos omaa edes pientä lajituntemusta, mutta hieman monimutkaisempien attribuuttien mukaan tuleminen saattaisi näyttää jo ennalta-arvaamattomia yhteyksiä. Mitä vahvempi yhteys attribuuteilla on, sitä pienempi hajonta pistepilveen tulee. Suorassa yhteydessä olevat attribuutit muodostavat viivan. Niin vahva korrelaatio johtuu usein vastakkaisesti tilastosta. Esimerkiksi kotijoukkueen tehdyt maalit ja vierasjoukkueen päästetyt maalit muodostaisivat lineaarisesti nousevan viivan, koska ne ovat saman tapahtuman eri suunnasta esitetyt tilastot. Eli aina kun kotijoukkue tekee 2 maalia, vierasjoukkue luonnollisesti päästää 2 maalia.

Korrelaatiossa olevat pisteet voivat asettua koordinaatistoon mukaillen joko nousevaa tai laskevaa janaa. Näiden lisäksi pisteet voivat mukailla käyrää tai paraabelia. On myös mahdollista, että pisteet muodostavat klustereita. Myös klusterit, eli keskittymät viittaavat siihen, että attribuuttien välillä on jotain yhteyksiä, ne eivät vain ole lineaarisessa suhteessa.

Tarkasteltaessa esimerkiksi kuvassa 6 kolmatta ruutua yläriviltä, pistepilven voi nähdä olevan kasvava. Kyseinen matriisin ruutu näyttää X-akselilla kotijoukkueen laukausten määrän maalia kohti ja Y-akselilla kotijoukkueen maalien määrän. Pisteet ovat pakkautuneet koordinaatiston oikeaan alakulmaan ihan siitä tilastollisesta syystä, että maaleja ei voi koskaan olla enemmän kuin laukauksia. Käytännössä tällainen voisi olla mahdollista, jos ottelussa tehdään oma maali, mutta johtuen tilastointimenettelystä jokaisesta maalista merkitään laukaus kohti maalia, oli peliväline toimitettu maaliin, miten tahansa. Kyseisessä tilastossa on havaittavissa sellainen mielenkiintoinen asia, että laukausten määrä ei suoraan korreloi maalien määrää. Toki pistepilvi on lineaarisesti nouseva ja otteluissa, joissa on tullut yli 10 laukausta, on sisältänyt vähintään yhden maalin, mutta koordinaatistosta on silti nähtävissä se, että laukausten määrä korreloi melko heikosti maalien määrään.

Tällaisesta matriisista voi myös havaita, jos datassa on jotain yllättäviä poikkeamia. Jos esimerkiksi yksittäinen piste näyttää asettuneet täysin poikkeukselliseen paikkaan, on syytä tarkistaa datasta kyseinen tapahtuma ja arvon oikeellisuus. (Minitab 18, 2019)

3 Neuroverkko (engl. Neural Network)

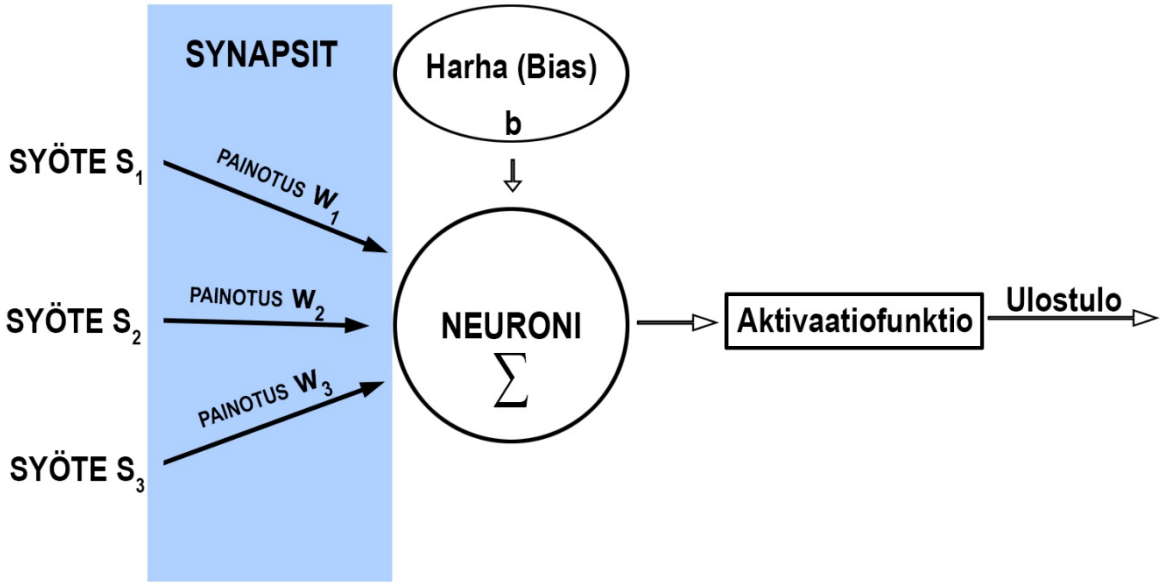
Neuroverkko on yksi suosituimmista koneoppimisen menetelmistä mahdollisesti monikäyttöisyytensä ansiosta. Neuroverkkoa käytetään mm. puheentunnistuksessa, tekstin syötön ennakoinnissa, tulevaisuuden ennustamisessa ja datan klusteroinnissa. Nimensä neuroverkko saa ihmisaivoja jäljittelevän rakenteensa ansiosta. (Cheung, 2020)

Neuroverkko koostuu neuroneista, jotka on asetettu omille tasoilleen. Tasot ovat asetettu rinnakkain. Neuronit ovat liitoksissa vierekkäisten tasojen neuroneihin synapsien avulla.

3.1 Neuroni

Neuroverkko koostuu neuroneista. Syötekerroksessa neuronit sisältävät neuroverkkoon syötetyn datan. Yksi syötekerroksen neuroni sisältää yhden attribuutin. Muut kuin syötekerroksen neuronit eivät lähtötilanteessa sisällä mitään tietoa vaan saavat sen vasta vastaanotettuaan sen aikaisemmalta neuronikerrokselta.

Kuva 7. Neuronin rakenne.



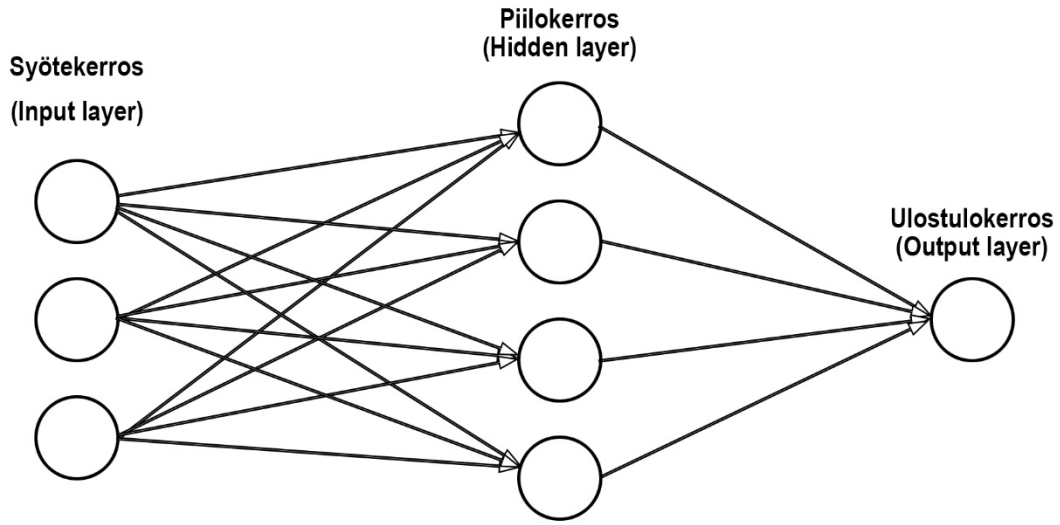
Neuronille syöte tulee synapsien välityksellä. Neuroneiden tarkoituksena on summata synapseilta tuleva tieto synapsien painotusten avulla. Painotuksen voi myös ajatella olevan

kyseisen asian tärkeys. Mitä relevantimpaa tieto on lopputuloksen kannalta, sitä suuremman painotuksen se saa. Neuroni kertoo syötteiden luvut painotuksilla, summaa ne ja lisää lopuksi siihen harhan (engl. bias). Laskukaava on siis $b + \sum_{i=1}^{N} s_1 w_1$. Harha on ylimääräinen osa neuroverkkoa, sen arvo on aina yksi, mutta sillä on oma painotuksensa. Harha ei ole pakollinen osa neuroverkkoa. Sen tarkoituksena on lopuksi korjata neuronin kokonaisarvoa, samaan tapaan kuin painotus korjaa yksittäisen syötteen arvoa. Ennen kuin neuroni on yhden iteraation osalta tehtävänsä suorittanut, se syöttää tiedon aktivaatiofunktioon. (Babs,2018)

3.2 Verkko

Neuroverkko koostuu tasoista. Ensimmäinen on syötekerros. Tällä tasolla on yhtä monta neuronia, kuin on lähtödatassa erilaisia signaaleja, eli arvoja.

Kuva 8. Neuroverkon rakenne.



Syötekerroksen jälkeen neuroverkossa on ensimmäinen piilokerros. Piilokerroksia voi yksi tai useampia. Jokainen piilokerroksen neuroni toteuttaa oman tehtävänsä. Verkon viimeinen osa on ulostulokerros. Ulostulokerroksen neuroni tai neuronit toimivat samaan tapaan kuin

piilokerroksen neuronitkin, mutta isoin ero tulee aktivaatiofunktiossa, jonka tarkoituksena on antaa lopullinen vastaus. (Logan, 2017)

3.3 Oppiminen

Neuroverkko oppii erehtymällä. Neuroverkko yrittää uudelleen useita eri variaatioita ja muuttaa omaa toimintaansa aina sen mukaan mikä antaa paremman lopputuloksen. Tarkkaan ottaen se saa ulostulolta tiedon virheen suuruudesta ja pyrkii pitämään tämän mahdollisimman pienenä. Validoinnin virhe (engl. validation loss) on saadun ja halutun lopputuleman erotus.

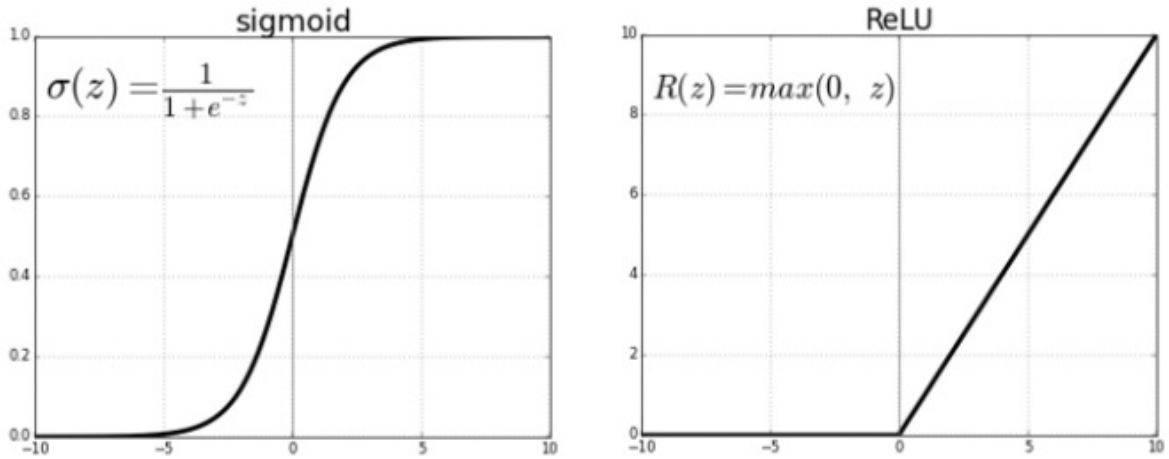
Tätä neuroverkon oppimismenetelmää voisi verrata leikkiin, jossa yksi etsii piilotettua esinettä ja muut huutavat ”kylmenee” ja ”lämpenee”, sen mukaan mihin suuntaan etsijä liikkuu. Tällä samalla tavalla neuroverkko kehittää itseään. Se aloittaa sattumanvaraisesta suunnasta ja vaihtaa suuntaa aina sinne missä ”lämpenee”. Neuroverkolle määritellään oppimisaste (engl. learning rate) jonka mukaan se vaihtaa suuntaa. Mitä suurempi oppimisaste, sitä suuremman muutoksen se tekee yritysten välissä. Käytännössä liian pieni arvo tekee etsinnästä hidasta, kun taas liian suuri tekee etsinnästä epäjohdonmukaista ja sattumanvaraista. Vertaa siihen, että joku huutaisi ”kylmenee” ja etsijä vaihtaisi suuntaa 0,1 astetta jatkaen lähes tismalleen alkuperäiseen suuntaan tai että hän kääntyisin aina 250 astetta ja poukkoilisi huoneessa sattumanvaraisesti.

Vastavirta-algoritmin (engl. backpropagation) tehtävä on kulkea verkko takaperin suorituksen jälkeen ja korjata matkalla painotuksia parhaan lopputuloksen saavuttamiseksi oppimisasteeksi määritellyn arvon suuruuden mukaan. (Kostadinov, 2019)

3.4 Aktivaatiofunktiot

Aktivaatiofunktion päätarkoitus on tehdä datalle epälineaarinen muunnos. Jos muunnosta ei tehtäisi, neuroverkko toimisi lineaarisesti, eikä voisi oppia vastavirta-algoritmin avulla.

Kuva 9. Aktivaatiofunktiot.



ReLU, Rectified linear unit on yleisimmin käytössä oleva aktivaatiofunktio neuroverkoissa. ReLU antaa arvon f(z) = 0, kun z on pienempi kuin nolla, mutta f(z) = z nollaa suuremmilla luvuilla. Ideana on antaa suoraan arvo nolla negatiivisille arvoille. Tämä parantaa neuroverkon toimintaa, kun seuraaville tasoille ei päästetä negatiivia arvoja.

Sigmoid-funktio on yleisin ulostulokerroksen aktivaatiofunktio binäärisissä luokitteluissa. Koska Sigmoid on logistinen funktio, sitä käytetään, kun luokkia on 2. Sigmoidista on myös päivitetty versio Tanh. Se on vastaava s-käyrä, mutta toisin kuin Sigmoid, se antaa arvon -1 ja 1 välillä.

Softmax aktivaatiofunktio luokittelee todennäköisyyden jokaiselle lopputulemalle. Lopputulosten summa on 1. Tätä käytetään ulostulokerroksissa, kun halutaan tietää todennäköisyys eri luokkien välillä tai kun luokkia on useita. (Brownlee, 2021)

4 Vahvuusluku

Useimmissa lajeissa on tarpeen sijoittaa kilpailijat järjestykseen taitotason mukaan. Tilanteissa, joissa halutaan pystyä kertomaan yksittäisen henkilön tai joukkueen taso suhteessa muihin, käytetään erilaisia vahvuuslukujärjestelmiä. Etenkin joukkueurheilussa nähdään käytettävän mallia, joka on todettu kaikista reiluimmaksi. Tässä Round Robin -

järjestelmässä kaikki kilpailijat pelaavat kaikkia vastaan ja korkeimman vahvuusluvun saa kilpailijat, joilla on eniten pisteitä. Tämän mallin ongelma on kuitenkin otteluiden suuri määrä. Yksilölajeissa tämän toteuttaminen olisi täysi mahdottomuus. Jotta nyrkkeilijän voisi kruunata maailman parhaaksi, Round Robin järjestelmää käyttäen, hänen tulisi kohdata jokainen maailman nyrkkeilijä.

Pienissä turnauksissa Round Robinin käyttö on mahdollista, eikä tämä ei aiheuta ongelmaa, mutta jo hieman suuremmissa turnauksissa kilpailijat täytyy asettaa pienempiin lohkoihin. Jos kilpailijat asetetaan lohkoihin täysin sattumanvaraisesti, saattaa käydä niin, että kaikki parhaat kilpailijat päätyvät samaan lohkoon ja näin ollen jatkopeleihin päätyy suhteettoman heikkoja kilpailijoita. Toki edelleen paras kilpailija turnauksen voittaa, mutta alemmista sijoista taisteltaisiin, hyvin isoin tasoeroin. (Sorensen, n.d.)

Jotta joukkueet jakaantuisivat mahdollisimman tasaväkisesti eri lohkoihin, esimerkiksi jalkapallon kansainvälisissä lopputurnauksissa on käytössä vahvuuslukujärjestelmä. Tällainen sijoitusjärjestelmä mahdollistaa joukkueiden sijoittamisen taitotason mukaan eri lohkoihin, vaikka kaikki mukana olevat joukkueet eivät ikinä olisi pelanneet toisiaan vastaan. Joukkueet päätyvät A, B, C, D ja E “koreihin” taitotason mukaan, kuhunkin 10 joukkuetta. Jokaiseen lohkoon sijoitetaan joukkue kustakin korista.

Jalkapallon kansainvälinen lajiliitto, FIFA (Fédération Internationale de Football Association), käyttää SUM nimistä järjestelmää joukkueiden luokitteluun. Jokaisen ottelun jälkeen lasketaan joukkueelle uusi vahvuusluku käyttäen seuraavaa kaavaa: $P = P_{before} + i \times (w - w_e)$.

P on uusi pistemäärä, eli vahvuusluku ja P_{before} on luku ennen ottelua. I on kerroin, joka määräytyy ottelun tärkeyden mukaan. Tällainen kerroin on tärkeä, koska esimerkiksi Suomi vastaan Espanja euroopanmestaruusfinaali on täysin eri asia, kun harjoituskaudella pelattu vastaava ottelu. I luku on pienimmillä 5 ja se annetaan, kun ottelu pelataan harjoitusotteluna virallisen kauden ulkopuolella. Suurimman kertoimen, eli painoarvon, 60, saa maailmanmestaruuskisojen välierästä ja finaalista, sekä mahdollisesta pronssiottelusta. (FIFA, 2018)

4.1 Elo-luku

Aikaisemmin esitelty SUM järjestelmä mukailee mahdollisesti kaikkein tunnetuinta luokittelujärjestelmää, Eloa. Elo Rating on alun perin shakkiin kehitetty luokittelujärjestelmä. Elon kehitti shakinpelaaja ja fysiikan professori Arpad Emrick Elo nimenomaan shakinpelaajien sijoitteluun. Alkuperäisesti Elo-luku muodostettiin kasvattamalla tai pienentämällä pelaajan Elo-lukua hänen saamiensa pisteiden perusteella. Tämä normaalijakaumaan perustuva malli korvattiin myöhemmin uudella versiolla, jossa ottelun odotettu Elo-lukujen muutos on pelaajien keskinäisten Elo-lukujen erotuksen logistinen funktio.

Elo-luku nykyisessä muodossaan lasketaan siis seuraavalla tavalla. Kunkin kilpailija saa aloitusluvun (yleisesti 1500) ja kun kilpailijat kohtaavat, heidän vanhat lukunsa päivitetään uusiin. Siihen kuinka paljon pisteitä voitosta saa tai kuinka paljon tappiosta menettää, riippuu kilpailijoiden tasoerosta ennen ottelua. Kilpailijoiden tasoero ilmoitetaan Elo-järjestelmässä odotusarvollisena lopputuloksena. Tämä on suhdeluku. Kilpailijoiden A ja B odotusarvollisten lopputulosten summa on 1.

Kilpailijat A odotusarvo kilpailijaa B vastaan lasketaan kaavalla $E_A = \frac{1}{1+10^{((R_B-R_A)/400)}}$.
Joukkueen A odotusarvollinen tulos E_A lasketaan kaavan 1 mukaisesti ja joukkueen B odotusarvollinen tulos E_B lasketaan vastavuoroisesti kääntämällä R_A ja R_B toisinpäin: $E_B = \frac{1}{1+10^{\left((R_A-R_B)/400\right)}}$. R_A ja R_B ovat joukkueiden A ja B Elo-luvut ennen ottelua.

Eli jos kilpailijan A Elo-luku on 1512 ja joukkueen B Elo-luku on 1624, joukkueen A voiton odotusarvo on 1 / (1 + 10^ ((1624-1512) / 400)) = ~0,3442 = ~34,4 % ja joukkueen B voiton odotusarvo on 1 / (1 + 10^ ((1512-1624) / 400)) = ~0,6558 = ~65,6 %.

Ottelun jälkeen Elo-luku kilpailijalle A päivitetään kaavan $R'_A = R_A + k(S_A - E_A)$ mukaisesti. R'_A on päivitetty arvo, R_A on vanha Elo-luku, k on K-luku, S_A on tulos (Voittaja 1 ja häviäjä 0. Tasapeli 0,5) ja E_A kilpailijan A voiton odotusarvo ennen ottelua.

K-luku on tärkeä osa järjestelmää. Sillä ilmoitetaan, kuinka vahva painoarvo ottelulla on. K-lukua suurentamalla luokituksen nousevat ja laskevat nopeammin, kun taas sitä pienentämällä luokitus ei muutu lyhyellä aikavälillä paljoakaan. K-lukua muuttamalla samaa järjestelmää voidaan käyttää useissa eri lajeissa ja sovelluksissa.

Eli jos aikaisemman esimerkin mukaisesti odotusarvollisesti parempi B voittaa ja käytämme K-lukua 20, A:n uusi Elo-luku on 1512 + 20 * (0 - ~0,3442) = ~1505 ja B:n uusi Elo-luku on 1624 + 20 * (1 - ~0,6558) = ~1631.

Jos samaisen ottelun voittaisikin odotusarvollisesti heikompi A, A:n uusi Elo-luku olisi ~1525 ja B:n uusi Elo-luku olisi ~1611. Huomionarvoista siis on, että odotusarvollisesti vahvempi kilpailija B voi voittaa 7 pistettä, mutta hävitä 13 pistettä. Ja koska kyseessä on nollasummajärjestelmä, kilpailija A voi vastavuoroisesti voittaa 13 pistettä, mutta hävitä vain 7 pistettä.

Laskukaava on siitä edistyksellinen, että tasatuloksessa pisteet eivät säily ennallaan, vaan tasapeli nostaa odotusarvollisesti heikomman pisteitä ja laskee vahvemman pisteitä. Tasapelin jälkeen pisteet olisivat A: 1621 ja B: 1515.

Koska kyseinen järjestelmä on kehitetty shakkiin, se ei ota huomioon kotietua. Kotietu kääntää kahden tasavahvan kilpailijan mittelön kotijoukkueen eduksi pitkässä juoksussa. Tutut pukuhuoneet ja ympäristö poistavat häiriötekijöitä ja helpottavat keskittymistä ja kotikentällä oikein sijoittuminen on helpompaa, kun mainostaulut yms. ovat tutuilla paikoilla. Isoin hyöty tulee kuitenkin psykologisista tekijöistä. Kotiyleisön kannustus nostaa itseluottamusta. Kotiyleisö aiheuttaa myös stressitekijöitä. Paineet voivat kasautua liian suuriksi ja tapahtuu ns. ylilatautuminen, joka heikentää suoritusta. (Iresearchnet, n.d.)

Tutkimusten mukaan myös tuomarit suosivat kotijoukkuetta, ilmeisesti juuri yleisön antaman paineen vuoksi. (Erikstad, 2020)

Kotiedun lisääminen Elo-järjestelmään tapahtuu lisäämällä kotijoukkueen ja vierasjoukkueen Elo-lukujen erotukseen kotietuarvon. Yleisemmin käytetty kotietu on 100 Elo-pistettä. Näin

ollen kaava kilpailijan A odotusarvon laskemiseksi, kun kilpailija A on kotijoukkue, $E_A = \frac{1}{1+10^{\left((R_B - R_A + 100)/400\right)}}$.

Koska kilpailijoilla on samasta ottelusta jaossa eri määrä pisteitä, Elo-järjestelmä on todettu toimimattomaksi osassa lajeissa. Ongelmalajeja ovat ne, joissa kilpailijat pääsevät valikoimaan vastustajiaan. Koska kyseessä on ns. nollasummajärjestelmä, tarkimpaan arvojärjestykseen päästään, vain jos osapuolet kohtaavat toisiaan sattumanvaraisesti. Kovassa nousussa olevaa heikompaa joukkuetta tai ottelijaa ei kannata arvoltaan korkealla olevan kohdata ollenkaan, koska voittamalla pisteitä on jaossa hyvin niukasti ja vastavuoroisesti häviöstä rangaistaan reilusti. (Sonas, 2020)

Toinen järjestelmän ongelmakohta johtuu deflaatiosta. Kun vanhat ja kokeneet kilpailijat lopettavat uransa huipulta, pisteitä katoaa järjestelmästä. Tämä luonnollisesti tapahtuu ainoastaan lähinnä yksilölajeissa. Nollasummapelin ekojärjestelmä pettää, kun käytössä olevien pisteiden arvo ei pysy samana. Kun esimerkiksi maailmanmestari lopettaa kilpailemisen, hänen 2300 pisteen saaliinsa poistuu ja muiden pisteet nostavat arvoaan. Keskiarvo ei tällöin enää olekaan 1500, vaan se on laskenut. Tällöin uusi kilpailija saa aloittaessaan kohtuuttoman korkean lähtötason. Järjestelmän kannalta tilanne ei ole mahdoton, mutta aiheuttaa ylläpidon kannalta päänvaivaa. Joukkue-urheilussa tilanne on helpommin hallittavissa, koska joukkueita ei katoa samalla tavalla. Myös inflaatio on mahdollinen. Jos sarjaan liittyy kilpailija, eikä kukaan poistu, keskiarvo nousee ja pisteet eivät ole enää niin arvokkaita. Tällöin uudelle kilpailijalle tulee antaa heti aluksi korkeampi Elo-luku, jossa inflaatio on otettu huomioon. (Silver, 2015)

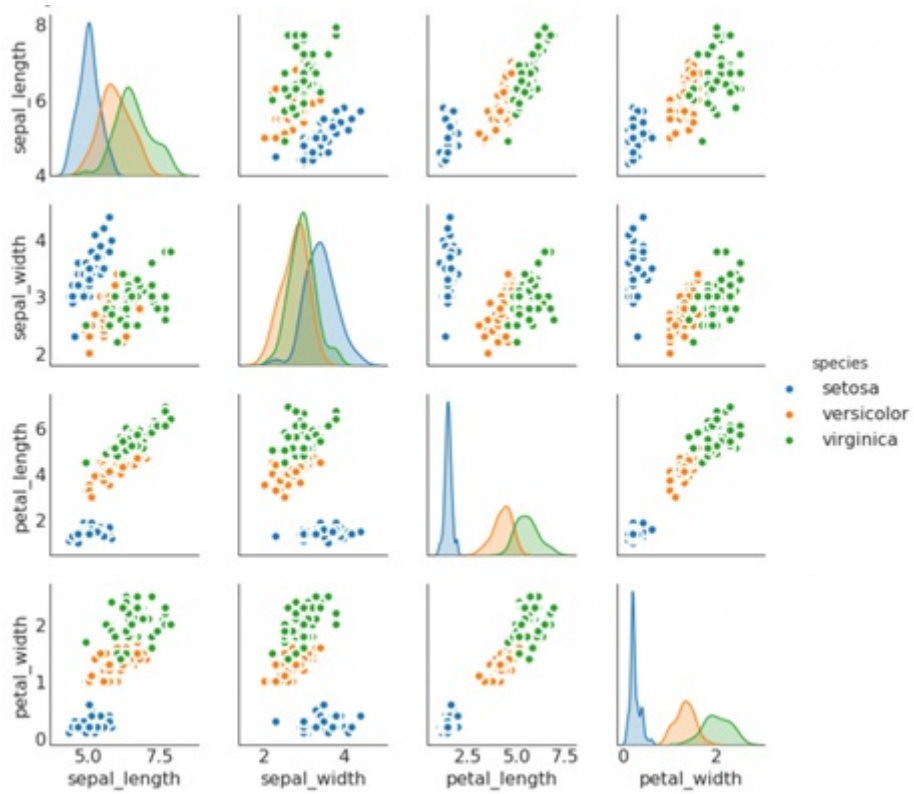
5 Python koneoppimisen ohjelmointikielenä

Koneoppimista voi ohjelmoida käytännössä millä tahansa ohjelmointikielellä, mutta Python on eniten käytetty. Pythonia käyttää noin 60 % ohjelmoijista. (Developer Nation, 2017)

Python on ohjelmointikielenä hyvin yleiskäyttöinen, mutta nimenomaan koneoppimisessa se on vakiinnuttanut vahvasti asemaa suosituimpana kielenä. Python ohjelmointikieli soveltuu

koneoppimiseen hyvin, sillä siihen löytyy koneoppimisessa käytettyjä hyödyllisiä kirjastoja. Tällaisia kirjastoja on muun muassa Numpy, jota käytetään matriisien ja kokoelmien (engl. array) tekoon, Pandas, joka on suorituskykyinen data analytiikassa sekä Matplotlib ja Seaborn, joilla saa hyvin yksinkertaisesti tehtyä erilaisia grafiikkoja datasta. (GeeksforGeeks, 2020)

Kuva 10. Matplotlibillä luotua grafiikkaa.



Näiden lisäksi Pythonille löytyy myös Sci-kit Learn, joka sisältää koneoppimisessa käytettyjä valmiita algoritmeja. Sci-kit Learn sisältää algoritmeja niin ohjattuun kuin ohjaamattomaan koneoppimiseen.

6 Opinnäytetyön tarkoitus, tavoite ja tutkimuskysymys

Elo-järjestelmä ja muut Elon pohjalta muunnellut sijoitusjärjestelmät kärsivät samasta ongelmasta, oikean k - kertoimen puuttumisesta. Kuten aiemmin todettu, k - kerroin

ilmaisee yksittäisen ottelun painoarvon. K luku itsessään on täysin sattumanvarainen luku ja oikean luvun löytää vain kokeilemalla.

Sijoitusjärjestelmän perusidea on laittaa kilpailijat paremmuusjärjestykseen. On monia syitä siihen, miksi kilpailijat halutaan järjestää paremmuuden mukaan, mutta enemmän tai vähemmän taustasyynä on se, että samantasoiset kilpailijat saisivat samanlaisen kohtelun. Joko kohtaamalla vain samantasoisia, kuten videopeleissä tehdään, tai kohtaamalla eri tasoisia, kuten kappaleen 4 esimerkissä esiteltiin. Urheiluvedonlyönnissä tai ylipäänsä ottelutulosten ennustamisessa samaa järjestelmää käytetään ikään kuin päinvastoin. Eli tilannetta lähestytään tuloksen suunnasta. Halutaan siis tietää myös seuraavan ottelun tulos. Menneisyys ei ole tulevaisuuden tae ja lopullinen tulos ratkeaa lukuisten asioiden summana. Sattuma, tai tuuri, näyttelee aina omaa osaansa. Se mitä tapahtuu ottelupäivänä tai sitä edeltävinä päivinä on usein erittäin ratkaisevaa ottelutuloksen kannalta, eikä joukkueen pitkäaikaisella menestyksellä ole lopulta juurikaan merkitystä.

Tällaisia asioita on mahdoton ennustaa pelkän sijoitusjärjestelmän perusteella. Selkeä esimerkki tähän on Aston Villan ja Liverpool FC:n kohtaaminen valioliigassa 8.1.2021. Yksi maailman suurimmista vedonvälittäjistä, Pinnacle, antoi edellisenä päivänä ottelun kertoimiksi 3,44 Aston Villan kotivoitto, 3,69 tasapeli ja 2,04 Liverpoolin vierasvoitto. Ottelupäivänä noin klo 10 Suomen aikaa tuli tieto, että osa Aston Villan joukkueen pelaajista olivat antaneet positiivisen näytteen Covid 19 - testissä ja näin ollen koko joukkue oli laitettu karanteeniin. Kertoimet muuttuivat tämän uutisen seurauksena rajusti ja olivat lopulta ottelun alkaessa 24,85, 15,17 ja 1,03. Kertoimet muutettuna prosenteiksi Aston Villan voiton todennäköisyys, vedonvälittäjän arvioissa, tippui 29,1 prosentista 4 prosenttiin. Liverpoolin voiton todennäköisyys nousi 49 %:sta 97,1 prosenttiin (Yhteenlaskettuna prosentit menevät yli sadan prosenttiyksikön johtuen vedonvälittäjän asettamasta katteesta).

Aston Villan joukkueessa pelasi lopulta täysin tuntemattomia seuran junioreita, jotka hävisivät kunniakkaasti ottelun, vain, 1–4. (Oddsportal, 2021)

Aina ottelun lähtöasetelma ei muutu näin räikeästi ja selvästi, vaan syyt ovat huomaamattomampia.

Yksittäisen ottelun ennustaminen vaatii siis paljon muutakin tietoa kilpailijoista, kuin vain heidän yleisen tasonsa, mutta ennustettaessa isompaa joukkoa otteluita, on erityisen tärkeää löytää jokin yhteinen nimittäjä sille, kumpi kilpailijoista voittaa. Jos sijoitusjärjestelmää käyttäen ottelun voittajaa ei voida ennustaa edes välttävästi, sijoitusjärjestelmää ei pitäisi käyttää mihinkään. Onhan sen lähtötarkoitus kuitenkaan osoittaa kilpailijoiden paremmuusjärjestys.

Tässä opinnäytetyössä on tarkoitus neuroverkon avulla poistaa k-luvun valitsemisen vaikeus, antamalla lähtödataksi korkean ja matalan k-luvun avulla laskettu Elo-luku. Tilannetta voisi lähestyä monella eri tavalla ja käyttäen muita koneoppimisen muotoja, mutta tähän työhön on valittu neuroverkko, koska se on todettu parhaiten toimivaksi koneoppimisen menetelmäksi nimenomaan urheilutulosten ennustamisessa.

Opinnäytetyön tarkoituksena on selvittää, neuroverkkoa (ANN) hyödyntäen, Elo -järjestelmän soveltuvuutta urheilutulosten ennustamiseen amerikkalaisen jalkapallon korkeimmalla tasolla (NFL).

Opinnäytetyö käsittelee yleisesti koneoppimista ja sen yleisimpiä käyttökohteita, sekä tarkastelee sekä yleisesti että neuroverkon avulla Elo-järjestelmän käyttöä toimivana sijoitusjärjestelmänä.

7 Opinnäytetyön toteutus

Opinnäytetyönä toteutettiin koneoppimismalli, joka oli vastavirta-algoritmia hyödyntävä neuroverkko (ANN), jonka tarkoituksena oli selvittää neuroverkon toimivuutta urheilutulosten ennustamisessa Elo-järjestelmää hyödyntäen.

7.1 Ohjelmointiympäristö ja -kieli

Koneoppimismalli rakennettiin Googlen Colaboratory -koneoppimisympäristössä. Colab soveltuu hyvin koneoppimismallin rakentamiseen, sillä koodin käsittely toteutetaan pienissä osioissa. Kun voidaan ajaa osissa, vikojen paikallistaminen nopeutuu. Tämä vähentää myös latausaikoja, kun koko koodia ei tarvitse suorittaa aina kerralla.

7.2 Datan valmistelu

Datana käytettiin valmiiksi strukturoitua csv-tiedostoa, joka sisälsi NFL-otteluiden tilastoja vuodesta 1966 lähtien. Koneoppimismallin luomisessa datan määrä on yksi tärkeimmistä lähtökohdista. Kun käytössä on tilasto lähes jokaisesta koskaan pelatusta ottelusta, voidaan hyvällä omalla tunnolla todeta, että dataa on riittävästi. Toki satunnaisuus näyttelee omaa osaansa, mutta koska vain ääretön määrä dataa antaisi meille äärettömän tarkkuuden, voimme olla tyytyväisiä tähän noin 16 000 ottelun tietokantaan. Tiedosto sisälsi otteluiden tulosten lisäksi ottelusijainnin, lämpötilan, tuulen ja ilmankosteuden.

Tiedostosta luodusta datakehyksestä (engl. dataframe) jätettiin pois ilmastoon liittyvät seikat, koska ne olisivat laajentaneet toteutusta merkittävästi. Tämä siitä syystä, että osassa otteluista käytössä on ollut katettu stadion. Joissain stadioneissa katto voidaan myös kesken ottelun siirtää paikalleen, joten ulkona vallitsevan säätilan vaikutusta siihen ilmastoon, jossa ottelu todellisuudessa on pelattu, vaatisi massiivista taustatutkimusta hallien rakenteiden tuntemisesta. Emme siis halua ottaa käyttöön sellaista mittausdataa, jonka luotettavuudesta ei voida olla varmoja.

Malliin haluttiin kaikista tärkeimmän elementin, eli luokitteluluvun lisäksi tuoda data lepopäivien määrästä. Tätä pidetään erittäin tärkeänä osana amerikkalaisessa jalkapallossa ja joidenkin arvioiden mukaan yli 6 päivän lepo otteluiden välissä nostaa voiton todennäkösyyttä merkittävästi. Koneoppimisen kauneus piileekin juuri siinä, ettei meidän tarvitse tietää onko tilastolla merkitystä lopulta vai ei, kun annamme koneen tehdä lopullisen päätöksen. Tätä tilastoa ei kuitenkaan suoraan ollut saatavilla, joten se jouduttiin laskemaan. Otteluiden päivämäärät olivat kuitenkin tiedossa, joten riitti että järjesti datan ensin joukkueittain järjestykseen ja sen jälkeen laski allekkain olevien päivämäärien erotuksen.

Datan muuntautuvuuskysymyksen äärelle päästiin, kun toteutettiin aikaisempaa joukkueiden järjestämistä nimen mukaan, sillä tässä kohtaa huomattiin, että joukkueiden nimet ovat vaihtuneet ajan saatossa. Muuntautunut data oli korjattavissa selvittämällä jokaisen nimeä vaihtaneen seuran historia. Osassa tapauksissa seura oli myyty tai siirretty toiseen kaupunkiin kuten esimerkiksi tapahtui Oakland Raidersille, kun se siirrettiin

Oaklandista Los Angelesiin vuonna 1982 ja joukkueen nimi vaihtui Los Angeles Raidersiksi. Liian pienen stadionin takia Los Angelesiin siirretty seura siirrettiin samasta syystä takaisin Oaklandiin vuonna 1995. Stadionia koskevat kiistat eivät kuitenkaan loppuneet vaan Raiders siirettii Las Vegasiin vuonna 2017. Omistaja ja seura kaiken kaikkiaan on siis pysynyt samana, joten dataa voidaan pitää luotettavana, kunhan muuntautuminen nimen suhteen korjataan. Datassa kaikki joukkueet nimettiin sen nimen mukaan, joka heillä oli käytössä kaudella 2020–2021. Joissain tapauksissa myös seuran nimi vaihtui kuten tapahtui Washington Redskinsille, kun rasisminvastaisen työn seurauksena sen nimi vaihdettiin, joidenkin mielestä jopa liiankin neutraaliksi, Washington Football Teamiksi.

Yleisesti arvostettu ja monin eri variaatioin hyödynnetty Elo – järjestelmä haluttiin tuoda mukaan yhdeksi muuttujaksi. Kuten k-lukua käsittelevässä kappaleessa kerrottiin, oikean k-luvun valitseminen on kuitenkin hyvin vaikeaa. K-luku 20 on yleisimmin käytetty luku joukkueurheilussa. Tämä luku ei kuitenkaan ota huomioon kovinkaan hyvin joukkueen tasoa lyhyellä aikavälillä, joten joukkueille laskettiin Elo – arvo myös k-luku 100:n mukaan. Joukkueille annettiin lähtöarvoksi 1500 ja tämän jälkeen jokainen ottelu laskettiin erikseen kahdesti. Toisella kerralla käytettiin k-lukua 20 ja toisella 100.

Lopuksi datasta siistittiin vielä pois alkupään rivit, joissa oli vajavaisia tietoja, kuten ensimmäiset ottelut, koska niissä ei luonnollisesti ollut lepopäiviä. Näin oli saatu kasaan Pythonin Pandas lisäosalla rakennettu datakehys (engl. dataframe), joka sisälsi 12 880 erillistä riviä. Yhdellä rivillä oli kotijoukkueen nimi, vierasjoukkueen nimi, kotijoukkueen ja vierasjoukkueen Elo-luku k = 20 ja Elo-luku k = 100, kotijoukkueen lepopäivät ja vierasjoukkueen lepopäivät sekä ottelun lopputulos. Yksi datakehyksen rivi voidaan ajatella yhtenä koneoppimismallin näytteenä (engl. sample).

7.3 Neuroverkon valmistelu

Jotta neuroverkon kehittämää mallia voitaisiin arvioida luotettavasti, sille ei anneta koko datajoukkoa käsiteltäväksi. Data oli siis jaettava kahtia niin, että 90 prosenttia siitä käytetään itse neuroverkon rakentamiseen ja loput 10 prosenttia jätetään neuroverkon testaamista varten. Näin vältytään siltä mahdollisuudelta, että neuroverkko oppisi oikeat vastaukset.

Testivaiheessa neuroverkko joutuu siis opittua mallia käyttäen ennustamaan ottelun tulokset vastaavan datan perusteella, tuntematta kuitenkaan oikeita vastauksia.

Koska datajoukko on kronologisessa järjestyksessä, järjestys tulee sekoittaa, jotta malli saa sattumanvaraisen otannan, eikä kohdenna mallin rakentamista johonkin tiettyyn ajanjaksoon. Näin ollen data sekoitettiin sattumanvaraiseen järjestykseen, ennen 9:1 kahtiajakoa.

Kuva 11. Ylimääräisistä tiedoista riisuttu datakehys.

elo_home_kuntopuntari	elo_away_kuntopuntari	elo_home_perustaso	elo_away_perustaso	rest_home_fix	rest_away_fix	result
1559.801377	1540.198623	1511.900474	1507.999526	7	8	1
1540.198623	1440.198623	1507.999526	1488.099526	5	5	1
1407.929697	1425.018522	1480.425402	1484.220210	7	9	1
1540.198623	1559.801377	1507.999526	1511.900474	7	8	2
1440.198623	1448.676521	1488.099526	1488.429326	15	8	2
1440.198623	1440.198623	1488.099526	1488.099526	7	8	2
1534.782855	1616.862696	1507.780264	1523.693291	9	8	1

Kun data on sekoitettu ja jaettu kahteen osaan, se standardisoitiin. Datan standardisointi tehdään siitä syystä, että datakehyksessä on täysin eri skaalassa olevia arvoja. Lepopäivät liikkuvat noin 4–14 päivän skaalassa, kun taas eloluokitukset 1500 yksikön molemmin puolin. Koska neuroverkko ei tiedä näiden numeroiden liittyvän eri asiaan, se ei voi myöskään antaa niille niiden ansaitsemaa suhteellista painoarvoa. 10 yksikön muutos lepopäivissä saattaa vaikuttaa lopputulokseen paljon enemmän kuin 10 yksikön muutos Elo-luokituksessa. Kuvassa 11 näkyy pieni otanta datasta ennen standardisointia ja kuvassa 12 standardisoinnin jälkeen.

Kuva 12. Standardisoitu datakehys valmiina mallin rakennukseen.

	0	1	2	3	4	5
0	0.322700	0.145459	0.090467	0.033439	-0.344506	0.068037
1	0.212381	-0.426757	0.050953	-0.171645	-1.120585	-1.137353
2	-0.531995	-0.513620	-0.228350	-0.211625	-0.344506	0.469834
3	0.212381	0.257629	0.050953	0.073642	-0.344506	0.068037
4	-0.350393	-0.378245	-0.150618	-0.168247	2.759812	0.068037
5	-0.350393	-0.426757	-0.150618	-0.171645	-0.344506	0.068037

Standardisointiin käytettiin Scikit Learn koodikirjaston StandardScaler nimistä työkalua.

Kuva 13. Ulostuloluokat binäärisessä muodossa.

	0	1	2	3	4	5	6	7	8	9	10	11
0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0
1	0.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0
2	0.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0
3	0.0	1.0	1.0	0.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0
4	0.0	1.0	1.0	0.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0
5	0.0	1.0	1.0	0.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0

Luokat (eng. Classes), eli tässä tapauksessa otteluiden tulokset (1, X, 2), tulee muuttaa binääriseen muotoon, koska toisin kuin regressiomallissa, tässä neuroverkossa tuloksia on vain kolme erilaista, eikä vastaus voi olla jotain näiden väliltä, eli kyseessä on luokittelusta (engl. classifier). Tähän käytettiin niin ikää valmista algoritmia, OneHotEncoderia. Kuvassa 13 näkyy luokat muutetussa muodossa. 3 ensimmäistä riviä on kotivoittoja ja 3 seuraavaa vierasvoittoja.

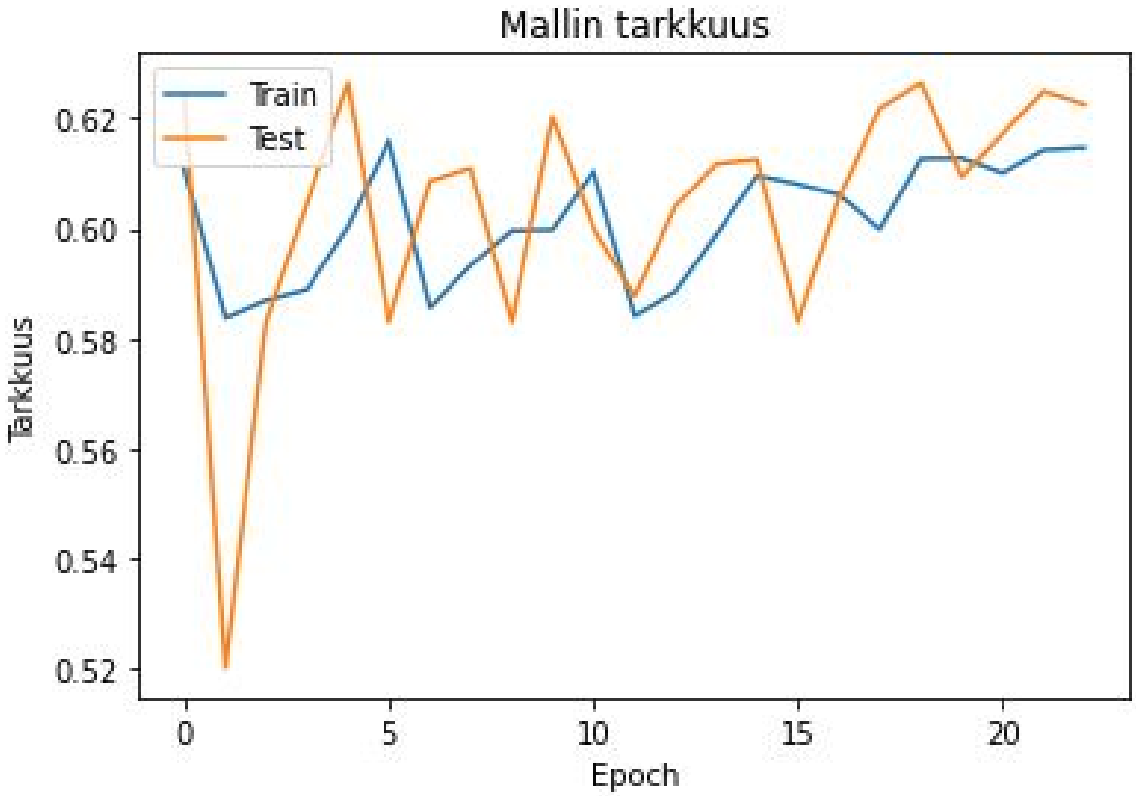
Mallin rakentamiseen päädyttiin käyttämään Keras-kirjastoa. Kun data on huolellisesti valmisteltu, Keraksen käynnistäminen on hyvin mutkaton toimenpide. Koska kyseessä on melko yksinkertainen datajoukko, liian monimutkaista neuroverkosta ei kannata tehdä. Esiasetuksista tärkein on tasojen (engl. layers) määrän ja neuroneiden määrän valitseminen. Tähän ei ole mitään laskukaavaa tai edes kunnollista nyrkkisääntöä. Parhaiten oikean verkon muodon löytyy vain kokeilemalla.

Machine Learning Mastery – verkkosivuston mukaan oikea neuroverkon muoto valitaan viidellä eri tavalla. Kaksi ensimmäistä ovat kokeileminen ja intuitio. Aloittelijalla ei välttämättä ole selkeää intuitiota oikeasta koosta, mutta muutaman kokeilun jälkeen asia alkaa hahmottua. Mallista kannattaa tulostaa aina grafiikka, joka näyttää mallin rakentamisen kehittyminen ja sen soveltuvuus testidataan. Tämä grafiikka antaa hyvää tietoa siitä, onko asetukset menossa oikeaan suuntaan. (Brownlee, 2019)

Kolmas hyvin oleellinen asetus on oppimisaste (engl. learning rate). Tämä määrittelee sen, kuinka isoja muutoksia malli tekee korjatessaan itseään. Mallin rakentumisen voi ajatella olevan tietokoneelle vastaava leikki kuin ihmisten leikkimä, kaikille tuttu, lämpenee-viilenee-leikki. Lähdet etsimään piilossa olevaa esinettä ja kun joku huutaa ”viilenee”, tiedät vaihtaa suuntaa. Ensimmäisellä kerralla teet täyskäännöksen, mutta jos silloinkin ”viilenee”, et ainakaan toivottavasti tee uutta täyskäännöstä ja lähde alkuperäiseen suuntaan vaan käännyt suuntaan, joka on jotain siltä väliltä. Hienosäätämällä suuntaa, johon kuljet, löydät piilotetun esineen. Tietokone etsii oikeaa mallia tällä samalla tavalla.

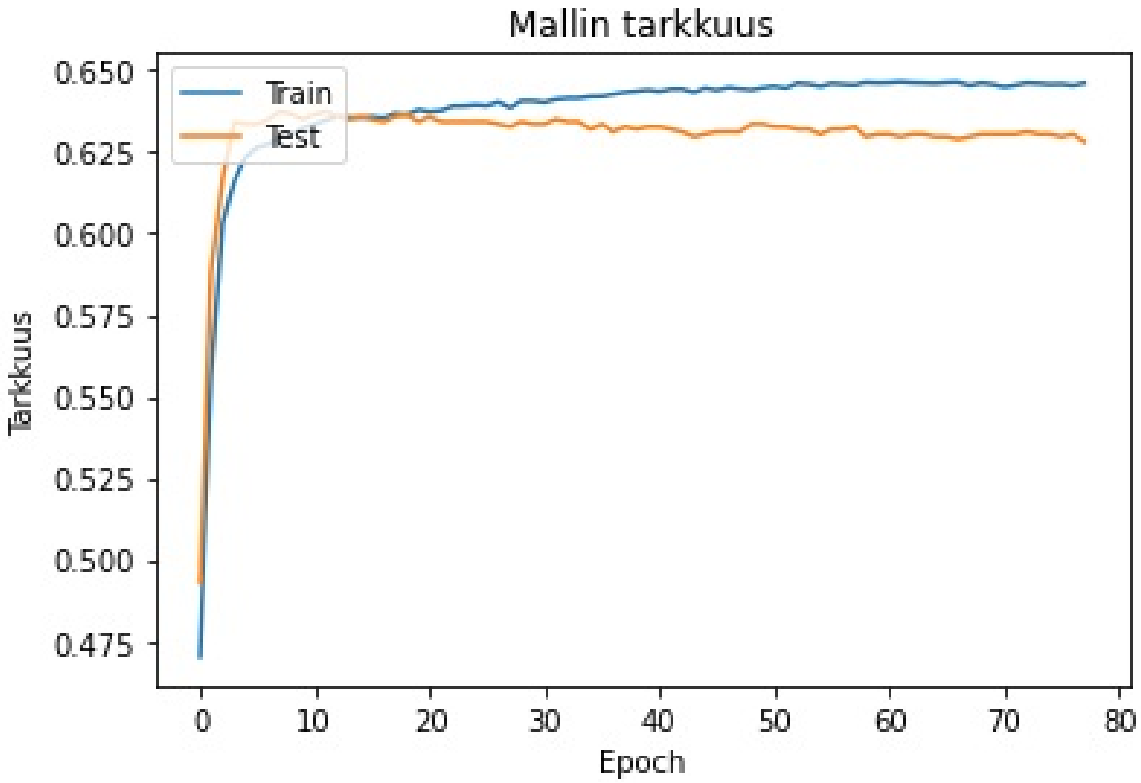
Neuroverkoilla on aina jokin sattumanvarainen lähtösuunta (engl. random seed). Alkuperäisellä lähtösuunnalla ei ole mitään merkitystä lopputuloksen kannalta, tärkeintä, että lähtee johonkin, koska vain paikallaan pysyminen tekee kohteen löytämisestä mahdotonta. Oikeaa mallia haarukoidessa onkin siis ihan normaalia, että joskus ensimmäinen epookki (engl. epoch) antaa tarkkuudeksi alle 20 %, mutta toisena päivänä saat, käyttämällä samaa mallia, jo ensimmäisessä mittauksessa hyvin lähelle lopullista arvoa olevan tarkkuuden - tietokone nyt vain sattumalta lähti etsimään oikeaa vastausta suoraan oikeasta paikasta.

Kuva 14. Neuroverkon oppimisprosessi liian korkealla oppimisasteella.



Jos oppimisaste on määritetty hyvin suureksi, kuten 0,1, tällöin tietokone poukkoilee laidasta laitaan etsiessään oikeaa suuntaa. Kuvassa 14 on esimerkki optimoinnin kehittymisestä 0,1:n oppimisasteella. Kuvassa 15 sen sijaan on 0,0001 oppimisaste ja grafiikasta on nähtävissä, kuinka malli lähenee hiljalleen kohti tasoa, josta se ei yksinkertaisesti pääse enää ohi. Keraksen Adam-algoritmi, joka vastaa tästä mallin optimoinnista, käyttää vakioasetuksena 0,001 oppimisastetta. Huomasimme kuitenkin, että pääsemme parempaan lopputulokseen pienentämällä tämän 0,0001:een.

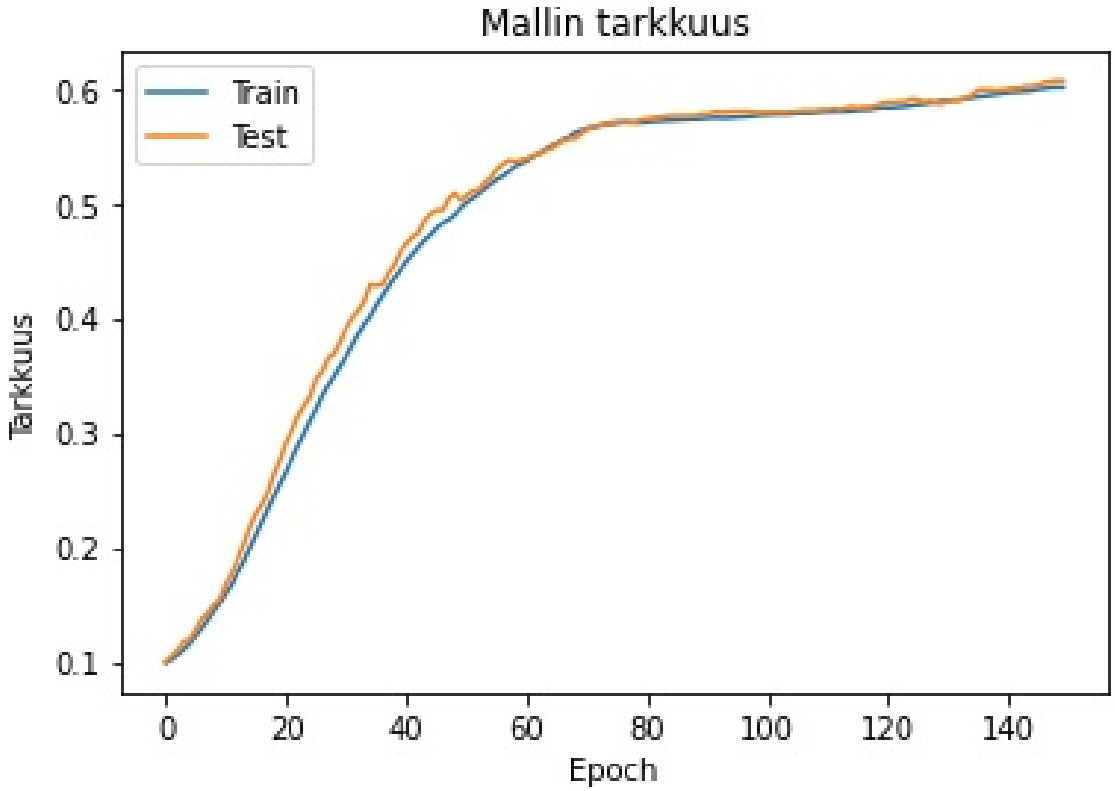
Kuva 15. Mallin tarkkuus 0.0001 oppisasteella.



Edellä mainittu epookki tulee myös itse määritellä. Tai tarkkaan ottaen niiden lukumäärä. Yksi epookki tarkoittaa datajoukon käymistä läpi yhden kerran. Tämä vaikuttaa siihen, kuinka kauan haluamme oikeaa mallia etsittävän. Koska oikean epookkimäärän valitseminen on melko turhauttavaa, opinnäytetyöhön rakennetussa mallissa otimme käyttöön Keraksen Callback-lisäosan. Tällä on mahdollista toteuttaa mallin tuottaminen niin, että kun malli ei enää kehity ja Keras saa 5:nnen (patience-luku 5 on vakioasetus.) kerran saman tarkkuuden, tietokone keskeyttää koodin ajon automaattisesti.

Opinnäytetyön koneoppimismalli pysähtyi 77 epookin jälkeen, eli koko datakehys käytiin läpi 77 kertaa. Tämä ei kuitenkaan tarkoita sitä, että neuroverkko olisi yrittänyt 77:ää erilaista mallia, vaan mallia korjataan aina jokaisen alijoukon (engl. mini batch) jälkeen. Mitä suurempi alijoukko valitaan, sitä tarkempia mallit ovat. Pienempi datajoukko antaa yksittäisen mallin nopeammin, mutta se on epätarkempi. Pienen alijoukon etu on taas se, ettei tietokone jää niin helposti ”jumiin” yhteen johtopäätökseen.

Kuva 16. Grafiikka hitaasta kehittymisestä.



Kuva 16 grafiikasta voi nähdä viitteitä mallin kehittymisestä ja siitä, kuinka sen kehitys hieman meinaa jumiutua, eli kehitys ei ole yhtä nopeasti kasvavaa, kun pienemmällä alijoukolla. Kuvassa on tulos, kun koneoppimismalliin kokeiltiin alijoukon kooksi 1000:n näytteen otantaa.

Käyttämällä samaa oppimisastetta kuin 32 näytteen alijoukolla, 1000 näytteen alijoukon tarkkuus jäi 3 prosenttiyksikköä heikommaksi. Muutettaessa alijoukon kokoa, tulee siis myös muuttaa oppimisastetta, koska ne ovat tärkeässä suhteessa keskenään. Samalla kuvasta voi huomata kuinka epookkien määrä kasvoi 117:ään. 1000 rivin alijoukolla kehitysasteen ollessa 0,00001 saamme mallin tarkkuudeksi vain 45,9 %, joka on todella heikko tulos. Parempaan ennustukseen pääsisi heittämällä kolikkoa. Suurentamalla oppimisaste 0,001:een tarkkuus nousi 65,45 %:iin, test-viiva ylitti train-viivan. Testitulos voi olla parempi kuin rakennusvaiheen tulos. Tämä johtuu siitä, että testivaiheessa yhden epookin aikana mallia kokeillaan usealla muodolla ja tulos on näiden keskiarvo. Paras malli kuitenkin valitaan

testidataan ja näin ollen on hyvin mahdollista, että testi näyttää antavan paremman tuloksen. (Keras, n.d.)

Taulukko 1. Oppimisasteen haarukointi 32 näytteen alijoukolla.

Oppimisaste	Tarkkuus	Epookkien lkm.
0,1	60.40372670807454	23
0,01	63.975155279503106	13
0,001	63.50931677018633	25
0,0001	64.44527950310559	104
0,00001	63.35403726708074	434

Taulukko 2. Alijoukon haarukointi 0,001 oppimisasteella.

Alijoukon koko	Tarkkuus
8	63.21739130434783
16	63.267080745341616
32	64.431677018633536
64	64.36335403726709
128	64.44099378881988
256	63.74223602484472
512	63.66459627329193

Oikeita asetuksia haettiin kokeilemalla ja lopulta päädyttiin käyttämään 0,0001 opetusastetta ja 32:n näytteen alijoukkoa.

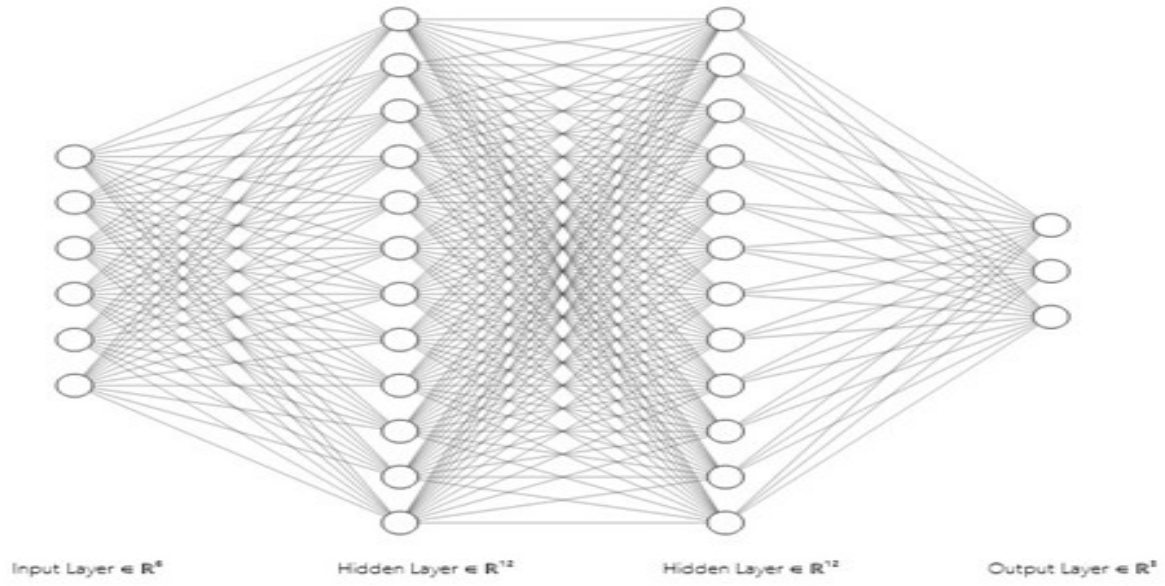
Laskentatapa siihen, kuinka monta erilaista koneoppimismallia neuroverkko yrittää kehittää yhden epookin aikana, saadaan jakamalla koko datajoukon rivien määrä alijoukon koolla. Opinnäytetyössä käytetty datakehys sisälsi 12 880 riviä ja siitä oli käytössä 90 % mallin kouluttamiseen. Alijoukon koon ollessa 32, koko datakehys jaetaan pienempiin osiin, niin että jokaiseen ryhmään tulee 32 näytettä. Eli yhden epookin aikana mallia kehitettiin, 12 880 * 0,9 / 32 = 362,25 kertaa. Koska jakolasku ei mene tasan Keras käyttää viimeiseen alijoukkoon loput jäljellejäävät. Eli lopulta yksi epookki sisälsi 363 erilaista mallia. Aikaisemmin esitelty callback funktio pysäytti laskennan 77 epookin jälkeen, eli lopulta tietokone oli yrittänyt 27 951 (77 epookkia * 363 alijoukkoa per epookki) erilaista mallia.

Tätä voi intuitiivisesti pitää isona määränä, koska käytössä oli datajoukko, jossa oli vain 6 erilaista muuttujaa. Kone kuitenkin sinnikkäästi muotoili niistä lähes 28 000 erilaista

kombinaatiota parhaan lopputuloksen saavuttamiseksi. Kuinka 6:sta eri muuttujasta voi saada 28 000 erilaista kombinaatiota, johtuu neuroverkon painotuksista. Kuten oli tarkemmin selitettynä kappaleessa 3.1, neuronit ovat yhteydessä toisiinsa keinotekoisilla synapseilla, joiden tehtävä on välittää dataa neuroneista toisiinsa. Neuronit ottavat synapseilta vastaan dataa erilaisin painotuksin. Johtuen näistä painotuksista 6 muuttujaa voidaan yhdistää toisiinsa loputtomalla määrällä eri variaatioita.

Puhekielessä verkon arkkitehtuuria kuvataan yleensä leveydellä ja syvyydellä. Nämä mitat voi ajatella mitattavan kulkusuunnan mukaan, eli leveä neuroverkko sisältää paljon neuroneita ja syvä verkko paljon eri tasoja.

Kuva 17. Neuroverkko graafisesti esitettynä.



Päädyimme käyttämään kahta piilotettua tasoa, joissa molemmissa oli 12 neuronia. Ensimmäinen taso oli luonnollisesti sisääntulotaso (engl. input layer) jossa oli 6 neuronia, johtuen siitä, että datakehyksessä on 6 eri syötettä (engl. input). Ulostulotaso (engl. output layer) sisälsi 3 neuronia, koska luokkia oli kolme (1, X ja 2). Ulostulotason aktivointi toteutettiin Softmax-menetelmällä. Softmax soveltuu täydellisesti juuri tämäntyyppiseen käyttötarkoitukseen.

Lopullinen malli onnistui ennustamaan otteluiden voittajat oikein 65,45 prosentin tarkkuudella.

Kuva 18. Kuvakaappaus tuloslistasta.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	team_home	team_away	elo_home_kuntopuntari	elo_away_kuntopuntari	elo_home_perustaso	elo_away_perustaso	rest_home_fix	rest_away_fix	result	prediction	pred_1	pred_x	pred_2	date
2811	Indianapolis Colts	Green Bay Packers	1562.9392337224695	1680.7144046205078	1498.8249897660392	1607.6552674711456	10	7	1	2	48.6	0.3	51.1	8.11.2020
2812	Jacksonville Jaguars	Pittsburgh Steelers	1132.2605153871673	1832.3482592498115	1339.3483936678356	1639.5473269038266	7	7	2	2	15.9	0.0	84.1	9.11.2020
2813	Las Vegas Raiders	Kansas City Chiefs	1580.1797429366297	1920.3658693978427	1466.7854142292606	1707.1976610045374	7	14	2	2	33.9	0.0	66.1	12.11.2020
2814	Los Angeles Chargers	New York Jets	1236.4639222203782	1221.2884819630235	1428.4194246079205	1349.9986344403646	7	13	1	1	59.6	0.3	40.1	15.11.2020
2815	Minnesota Vikings	Dallas Cowboys	1572.2606411602044	1290.835190261868	1550.4663627230084	1490.8671188206538	6	14	2	1	74.9	0.2	24.9	15.11.2020
2816	New Orleans Saints	Atlanta Falcons	1758.0644088854874	1485.665886185887	1658.4550535781163	1495.8330955542594	7	14	1	1	77.5	0.1	22.4	15.11.2020
2817	Washington Football Team	Cincinnati Bengals	1153.0872186285653	1265.2721407379906	1353.9608384891285	1358.8495069704536	7	7	1	1	51.8	0.2	47.9	15.11.2020
2818	Tampa Bay Buccaneers	Los Angeles Rams	1592.0298420707488	1613.990254560918	1467.651097940666	1574.6360129355164	8	8	2	1	57.4	0.4	42.3	15.11.2020
2819	Dallas Cowboys	Washington Football Team	1368.0911196823836	1227.0297595202424	1500.5479618685965	1365.9955570508087	4	4	2	1	69.8	0.2	30.0	15.11.2020
2820	Detroit Lions	Houston Texans	1272.4427542056887	1499.330420122517	1396.4045758258362	1499.619912450643	4	4	2	2	44.6	0.4	54.9	15.11.2020
2821	Pittsburgh Steelers	Baltimore Ravens	1833.5289131769064	1658.0137987420005	1641.6699389167916	1611.4287538585156	4	4	1	1	70.7	0.1	29.2	15.11.2020
2822	Atlanta Falcons	Las Vegas Raiders	1461.9958571603477	1571.5165040358054	1488.5012147511911	1463.9165330468759	7	7	1	1	50.2	1.2	48.6	15.11.2020
2823	Buffalo Bills	Los Angeles Chargers	1629.7589417007516	1294.1485202211811	1531.782738148327	1438.0996943328473	14	7	1	1	77.5	0.1	22.3	15.11.2020
2824	Cincinnati Bengals	New York Giants	1191.3295998463134	1347.6495848465327	1346.8147884087728	1367.753383122936	7	14	2	2	49.3	0.3	50.4	15.11.2020
2825	Denver Broncos	New Orleans Saints	1492.6975637200594	1781.7344379110268	1473.9307943534266	1665.7869343811847	7	7	2	2	37.4	0.1	62.5	15.11.2020
2826	Green Bay Packers	Chicago Bears	1606.1566262482943	1451.9270094441101	1593.0150860455558	1490.1344259398855	7	13	1	1	70.2	0.3	29.5	16.11.2020
2827	Indianapolis Colts	Tennessee Titans	1637.497012094683	1685.2249495466224	1513.465171191629	1551.3671116271246	7	7	2	1	57.5	0.3	42.2	19.11.2020
2828	Jacksonville Jaguars	Cleveland Browns	1131.079861460072	1573.5116853629404	1337.22578165487	1414.2248047274632	7	7	2	2	36.4	0.1	63.4	22.11.2020
2829	Los Angeles Rams	San Francisco 49ers	1651.1910976280453	1529.0751113444458	1579.937039950122	1513.3087554777787	6	14	2	1	67.3	0.4	32.2	22.11.2020
2830	Minnesota Vikings	Carolina Panthers	1495.0047117396894	1363.3901973212746	1540.7855196750656	1446.5676386402338	7	7	1	1	68.2	0.4	31.3	22.11.2020
2831	New England Patriots	Arizona Cardinals	1494.6834503688906	1524.242501248441	1649.5049034480967	1461.6000066443028	7	10	1	1	69.0	0.6	30.5	22.11.2020
2832	New York Jets	Miami Dolphins	1163.6038839622208	1530.1313669883411	1340.3183647154374	1464.8216983185382	7	7	2	2	37.7	0.2	62.1	22.11.2020
2833	Tampa Bay Buccaneers	Kansas City Chiefs	1554.8289990036214	1929.029108298667	1462.3500709260604	1710.0665421869214	6	7	2	2	34.0	0.0	66.0	22.11.2020
2834	Philadelphia Eagles	Seattle Seahawks	1403.2816789072995	1646.8729741405843	1525.9281324584335	1613.1261658557598	8	11	2	2	39.3	0.4	60.2	22.11.2020
2835	Baltimore Ravens	Dallas Cowboys	1622.8790777127576	1307.866848860677	1600.3733451089277	1488.742609436684	7	7	1	1	77.8	0.1	22.1	22.11.2020
2836	Arizona Cardinals	Los Angeles Rams	1460.4273188441193	1593.606093923321	1454.9282526849795	1570.0549610380408	7	7	2	2	46.9	0.6	52.5	22.11.2020
2837	Atlanta Falcons	New Orleans Saints	1535.6418122552109	1793.028738431962	1499.7163143864504	1669.4124302516068	7	7	2	2	40.5	0.1	59.4	22.11.2020
2838	Chicago Bears	Detroit Lions	1413.9522834176873	1257.0388396070048	1481.1516499282807	1391.0187347648725	7	10	2	2	69.3	0.3	30.4	22.11.2020
2839	Green Bay Packers	Philadelphia Eagles	1644.1313522747168	1389.0895123721	1601.99786205713	1520.1725433648794	7	6	1	1	74.4	0.2	25.4	22.11.2020
2840	Houston Texans	Indianapolis Colts	1514.7343347212006	1603.6907547281853	1505.0057535116068	1506.4835825282282	10	7	2	1	52.9	1.2	45.9	23.11.2020
2841	Kansas City Chiefs	Denver Broncos	1936.2632636368194	1481.4032631991245	1712.8337236344275	1470.3052984830044	7	7	1	1	85.8	0.0	14.2	26.11.2020
2842	Los Angeles Chargers	New England Patriots	1276.4182949209946	1558.4986327732122	1428.8553904073613	1656.1766574074195	7	7	2	2	30.6	0.2	69.2	26.11.2020
2843	Miami Dolphins	Cincinnati Bengals	1537.6675768509122	1169.8633424650254	1469.7398938482154	1339.384454428543	7	7	1	1	79.6	0.1	20.4	26.11.2020
2844	Minnesota Vikings	Jacksonville Jaguars	1536.090063127917	1126.0779575796132	1550.0145854406378	1331.2270747550556	7	7	1	1	82.2	0.0	17.7	29.11.2020
2845	New York Jets	Las Vegas Raiders	1156.0676740996496	1497.8705489409424	1335.4001691857604	1452.7014334116166	7	7	2	2	39.3	0.2	60.6	29.11.2020

Lopuksi ennustukset sisältävään datakehykseen palautettiin joukkueiden nimet, jotta sen lukeminen olisi informatiivisempaa.

8 Johtopäätökset ja pohdinta

Neuroverkon voi hyvällä omalla tunnolla todeta olevan loistava työkalu urheilutulosten ennustamiseen. Tärkeimpänä lähtöarvona käytetty Elo-järjestelmä antaa hyvät lähtötiedot neuroverkolle. 65,45 % osumatarkkuus on riittävä siihen, että voidaan sanoa, että lähtötiedoilla ja tuloksella on selkeä korrelaatio, ja neuroverkko onnistui sen löytämään. Mistään merkittävästä tuloksesta ei voida kuitenkaan puhua sillä, jos asiaa tarkastelisi esimerkiksi vedonlyönnin näkökulmasta, 65,45 % tuskin riittäisi voitolliseen vedonlyöntiin.

Neuroverkon hienous on siinä, että sitä voi aina jatkokehittää antamalla sille jotain lisätietoja. Jokainen pienikin tiedonjyvänen mahdollistaa parempien ennustusten laatimisen. Eivätkä ne välttämättä vaadi merkittäviä muutoksia itse mallin arkkitehtuuriin.

Neuroverkko käyttö tulee varmasti lisääntymään, sillä se ei teknologiana vaadi kovin suurta tietoteknistä osaamista. Datan valmistelu sen sijaan on pitkä ja monivaiheinen prosessi toteuttaa ja tässä opinnäytetyössä se tehtiin täysin ohjelmoimalla, mutta välttävillä Excel taidoilla pystyisi toteuttamaan saman asian.

Lähteet

ABCadda. 2020. One of the pillar of big data: Semi structured Data. Haettu 30.6.2021 osoitteesta https://abcadda.com/one-of-the-pillar-of-big-data-semi-structured-data/

Algorithm-X Lab: 10 Use Cases of Neural Networks in Business. Haettu 1.8.2021 osoitteesta https://algorithmxlab.com/blog/10-use-cases-neural-networks/#What_are_Artificial_Neural_Networks_Used_for

Babs, T. 2018. The Mathematics of Neural Networks. Haettu 16.8.2021 osoitteesta https://medium.com/coinmonks/the-mathematics-of-neural-network-60a112dd3e05

BBVA. 2020. The five V's of big data. Haettu 30.6.2021 osoitteesta https://www.bbva.com/en/five-vs-big-data/

Brownlee, J. 2018. Machine Learning Mastery: How to Configure the Number of Layers and Nodes in a Neural Network. 6.8.2019. Haettu 24.11.2020 osoitteesta https://machinelearningmastery.com/how-to-configure-the-number-of-layers-and-nodes-in-a-neural-network/

Brownlee, J. 2021. How to Choose an Activation Function for Deep Learning. Haettu 16.8.2021 osoitteesta https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/

Curtis, B. (n.d.). What are the 7 V's of Big Data? Haettu 22.7.2021 osoitteesta https://www.yourtechdiet.com/blogs/7vs-big-data/#Variability

Developer Nation. 2017. Towards Data Science: What is the best programming language for Machine Learning? 5.5.2017. Haettu 6.7.2020 osoitteesta https://towardsdatascience.com/what-is-the-best-programming-language-for-machine-learning-a745c156d6b7

Elements of AI. (n.d.). Koneoppimisen lajit. Haettu 12.11.2020 osoitteesta https://course.elementsofai.com/fi/4/1

Erikstad, M. 2020. Referee Bias in Professional Football: Favoritism Toward Successful Teams in Potential Penalty Situations. 27.2.2020. Haettu 22.2.2021 osoitteesta https://www.frontiersin.org/articles/10.3389/fspor.2020.00019/full

European Business Review. 2021. Haettu 1.5.2021 osoitteesta https://www.europeanbusinessreview.com/can-ai-make-winning-sports-betting-picks/

eWeek. 2019. How Data Itself Will Take IT Business to a New Level. https://www.eweek.com/big-data-and-analytics/how-data-itself-will-take-it-business-to-a-new-level/

FIFA. 2018. Revision of the FIFA / Coca-Cola World Ranking. Haettu 7.1.2021 osoitteesta https://digitalhub.fifa.com/m/f99da4f73212220/original/edbm045h0udbwkqew35a-pdf.pdf

Geeks for Geeks. 2020. Top 5 Programming Languages and their Libraries for Machine Learning in 2020. Haettu 14.11.2020 osoitteesta https://www.geeksforgeeks.org/top-5-programming-languages-and-their-libraries-for-machine-learning-in-2020/

Guru99.(n.d.). Guru99: Reinforcement Learning: What is, Algorithms, Applications, Example. Haettu 16.8.2021 osoitteesta https://www.guru99.com/reinforcement-learning-tutorial.html

Huston, I. 2019. How AI and Software 2.0 will change the role of programmers. 30.5.2019. Haettu 20.9.2020 osoitteesta https://bdtechtalks.com/2019/05/30/ai-software-2-automated-programming/

IBM. (n.d.). Unsupervised learning. Haettu 16.8.2021 osoitteesta https://www.ibm.com/cloud/learn/unsupervised-learning

Impact. 2016. The 7 V's of Big Data. Haettu 30.6.2021 osoitteesta https://impact.com/marketing-intelligence/7-vs-big-data/

International Data Corporation. (2020). IDC's Global DataSphere Forecast Shows Continued Steady Growth in the Creation and Consumption of Data. Noudettu osoitteesta https://www.idc.com/getdoc.jsp?containerId=prUS46286020

Iresearchnet. (n.d.). Home-Field Advantage. Haettu 15.8.2021 osoitteesta http://psychology.iresearchnet.com/social-psychology/control/home-field-advantage/

Keras. (n.d.). Keras FAQ: Why is my training loss much higher than my testing loss? Haettu 23.6.2021 osoitteesta https://keras.io/getting_started/faq/#why-is-my-training-loss-much-higher-than-my-testing-loss

Kokkarinen, I. & Ala-Mutka K. (2002). Tietokoneet ja algoritmit. Helsinki: Talentum Media Oy.

Kostadinov, S. 2019. Understanding Backpropagation Algorithm. Haettu 16.8.2021 osoitteesta https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd

LI, H. 2020. SAS: Which machine learning algorithm should I use? Haettu 16.8.2021 osoitteesta https://blogs.sas.com/content/subconsciousmusings/2020/12/09/machine-learning-algorithm-use/

Logan, S. 2017. Understanding the Structure of Neural Network. Haettu 16.8.2021 osoitteesta https://becominghuman.ai/understanding-the-structure-of-neural-networks-1fa5bd17fef0

Mears, B. 2015. Fantasy Labs: How Much Does Days Rest Affect Each NFL DFS Position? 20.7.2015. Haettu 15.8.2021 osoitteesta https://www.fantasylabs.com/articles/how-much-does-days-rest-affect-each-nfl-dfs-position/

Minitab 18 Support. 2019. Interpret the key results for Matrix Plot. Haettu 14.9.2020 osoitteesta https://support.minitab.com/en-us/minitab/18/help-and-how-to/graphs/how-to/matrix-plot/interpret-the-results/key-results/#step-1-look-for-model-relationships-and-assess-the-strength

Oddsportal. 2021. Aston Villa – Liverpool. Haettu 10.1.2021 osoitteesta https://www.oddsportal.com/soccer/england/fa-cup/aston-villa-liverpool-UmbYy9RI/

Salo, I. 2014. Big Data ja Pilvipalvelut. Docento Oy.Cheung, K. 2020.

Shapiro, R & Aneja S. (n. d.). Who Owns Americans' Personal Information and What Is It Worth? Noudettu osoitteesta https://futuremajority.org/wp-content/uploads/PersonalInfo.pdf

Silver, N. & Fischer-Baum, R. 2015. FiveThirtyEight: How We Calculate NBA Elo Ratings. 21.5.2015. Haettu 10.7.2021 osoitteesta https://fivethirtyeight.com/features/how-we-calculate-nba-elo-ratings/

Singhal, R. 2019. Programming a Paradigm Shift: Software 1.0 to Software 2.0. 2.8.2019. Haettu 1.1.2020 osoitteesta https://www.tcs.com/blogs/programming-a-paradigm-shift-software

Sonas, J. 2020. What's Wrong With the Elo System? 19.4.2020. Haettu 2.3.2020 osoitteesta https://en.chessbase.com/post/what-s-wrong-with-the-elo-system

Sorensen, S. (n.d.). An overview of some methods for ranking sports teams. Haettu 2.2.2021 osoitteesta http://www.phys.utk.edu/sorensen/ranking/Documentation/Sorensen_documentation_v1.pdf

Track and Field Statistics. 2004. Haettu 22.7.2021 osoitteesta http://trackfield.brinkster.net/RecProg_All.asp?RecCode=WR&EventCode=MF8&Gender=M&P=F

van Rijmenam, M. 2013. A Short History Of Big Data. 7.1.2013. Haettu 16.5.2021 osoitteesta https://dataflog.com/read/big-data-history/239

Wiele, C. 2019. AI for Everyone: Why We Need Machine Learning. 16.1.2019. Haettu 3.6.2020 osoitteesta https://medium.datadriveninvestor.com/ai-for-everyone-why-we-need-machine-learning-81de7b6b7f64

Wolff, R. 2020. 5 Types of Classification Algorithms in Machine Learning. 26.8.2020. Haettu 5.12.2020 osoitteesta https://monkeylearn.com/blog/classification-algorithms/