



TEKNILLINEN TIEDEKUNTA

# **Keinotekoiset neuroverkot koneoppimisessä**

Kristian Tiiro

PROSESSITEKNIikka

Kandidaatintyö

Maaliskuu 2020

# TIIVISTELMÄ

Keinotekoiset neuroverkot koneoppimisessa

Kristian Tiiro

Oulun yliopisto, Prosessitekniiikan tutkinto-ohjelma

Kandidaatintyö 2020, 34 s.

Työn ohjaajat yliopistolla: TkT Aki Sorsa, TkT Ari Isokangas

Työn tarkoituksena on antaa lukijalle käsitys koneoppimisesta, keinotekoisista neuroverkoista ja syväoppimisesta. Työssä käsitteet määritellään ja niitä käsitellään lähinnä kuvailevalla tasolla menemättä syvälle matemaattisiin kuvauksiin ja malleihin. Tieto perustuu alan teoksiin. Lisäksi tarkastellaan keinotekoisien neuroverkkojen nykyisiä sovelluksia, yhteiskunnallisia vaikutuksia ja mahdollisia tulevaisuudennäkymiä. Tiivistettynä voidaan ilmaista, että keinotekoisin neuroverkoin toteutettu syväoppiminen on nykyaikana nouseva koneoppimisen haara, joka ei kuitenkaan ole verrattavissa biologisiin aivoihin perusidea syvemmällä eikä omaa tietoisuutta. Keinotekoisien neuroverkon toimintaperiaate on alkeellisella tasolla ymmärrettävissä, jopa ilman suurta tietämystä tietotekniikasta tai isoa määrää matemaattisia malleja.

*Asiasanat: tekoäly, koneoppiminen, keinotekoiset neuroverkot, syväoppiminen*

# SISÄLLYSLUETTELO

1 Johdanto .....	5
2 Koneoppiminen .....	7
2.1 Johdatus koneoppimiseen.....	7
2.2 Koneoppimisen tekniikat .....	8
2.3 Koneoppimisen lähestymistavat.....	8
2.3.1 Ohjattu oppiminen .....	9
2.3.2 Ohjaamaton oppiminen.....	9
2.3.3 Vahvistusoppiminen .....	9
3 Keinotekoiset neuroverkot ja syväoppiminen.....	11
3.1 Mitä on syväoppiminen? .....	11
3.2 Biologisten aivojen toimintaperiaate.....	12
3.3 Neuronit.....	14
3.3.1 Aktivaatiofunktio .....	15
3.3.2 Stokastiset mallit.....	16
3.4 Neuroverkon kuvaaminen kaaviolla .....	17
3.5 Verkon rakenne .....	18
3.5.1 Monikerroksiset eteenpäin kytketyt verkot .....	18
3.5.2 Recurrent-verkot .....	20
3.6 Tiedon esittäminen verkolle.....	21
3.7 Oppiminen.....	23
3.7.1 Backpropagation-algoritmi .....	23
3.7.2 Ylioppiminen .....	25
3.8 Käyttökohteita .....	26
4 Pohdintoja tulevaisuudenkuvista.....	29
5 Yhteenveto .....	32
6 Lähdeluettelo.....	33

## MERKINNÄT JA LYHENTEET

$C$	virhefunktio
$P(v_k)$	stokastisessa aktivaatiofunktiossa todennäköisyys ulostulon arvolle 1
$a$	aktivaatiofunktioissa esiintyvä funktion jyrkkyyteen vaikuttava parametri
$a^*$	vektori ulostulokerroksen neuroneiden toteutuneista ulostuloista
$b_k$	vakiotermin neuronille $k$ (bias engl.)
$b^*$	vektori, joka edustaa kaikkia verkon vakiotermejä (bias engl.)
$n$	opetuserän esimerkkien määrä
$v_k$	neuronin $k$ indusoitunut paikallinen kenttä
$w_{kj}$	neuronin $k$ painotuskerroin yhteydelle $j$
$w^*$	vektori, joka edustaa kaikkia verkon painotuskertoimia
$x_j$	neuroniin saapuva tulosignaali yhteydestä $j$
$x^*$	vektori yksittäisen opetusesimerkin tulosignaaleista
$y_k$	neuronin $k$ ulostulo
$y^*$	vektori halutuista viimeisen kerroksen neuroneiden ulostuloista
$z^{-1}$	Yhden aikayksikön suuruinen aikaviive takaisinkytketyssä piirissä
$\varphi(v_k)$	neuronin $k$ aktivaatiofunktio

# 1 JOHDANTO

Voidaan ajatella, että tekoälyn tuomat teknologiset mahdollisuudet ovat puhuttaneet ihmisiä jo antiikin ajoista lähtien, jolloin muinaiset kreikkalaiset pohtivat mahdollisuutta siirtää ihmismieli mekaaniseen koneeseen muodostaen mekaanisen ihmisen, jota kutsuttiin myyteissä Talokseksi. Kreikkalaiset eivät olleet ainoa laatuaan vastaavien ajatusten kanssa, vaan myytti juutalaisten Golemista pyörii myös mekaanisen, mutta ajattelevan henkilön ympärillä. Jopa eurooppalaiset valistusajan filosofit, kuten René Descartes, pohtivat mahdollisuutta kuvata kaikkea ajattelua matemaattisesti. (Mueller & Massaron, 2017, s.12)

Nykyään ihmiskunta on jo ainakin huomattavasti lähempänä itsenäisesti ajattelevaa konetta, ja erityisesti viime vuosikymmeninä tekoälyn tuomat mahdollisuudet ja uhkakuvat ovat usein näkyneet populäärikulttuurissa, kuten sci-fi elokuvissa ja kirjoissa, joista voidaan mainita esimerkiksi 2001 Space Odysseyn HAL tai Star Warsin C-3PO. Kuitenkin monet teknologiset asiat, joita ennen nähtiin elokuvissa ja pidettiin naurettavina ovat sitten tulleet toteen, kuten kosketusnäytöt ja videopuhelut, mikä voi herättää kysymyksiä myös tekoälyn edistyksestä. Mitä tekoälyllä voidaan nykyään ja lähitulevaisuudessa saada aikaan ja mitä tarkoitetaan koneoppimisella? Mihin sovellukset perustuvat ja kuinka kehittyneitä ja itsenäisiä nykyiset tekoälymme ovat? Mitä tarkoitetaan, kun puhutaan aivojen inspiroimista keinoitekoisista neuroverkoista ja olisiko Skynet Terminatorista mahdollinen?

Koneoppiminen ja erityisesti keinoitekoisten neuroverkoin toteutettu syväoppiminen ovat nykyaikana usein kuultuja termejä, ja ne ovat jatkuvasti enemmän läsnä jokapäiväisessä arjessa, vaikka älypuhelin ja sosiaalisen median kautta. Esimerkkinä siitä on valokuvien kasvojentunnistus Facebookissa tai puheentunnistus älypuhelimien hakutoiminnoissa. Harvalla ihmisistä on silti juuri mitään käsitystä, mihin kyseiset tekoälyn alle sijoittuvat sovellukset perustuvat ja mihin kaikkeen niitä voidaan hyödyntää. Tämän työn tarkoituksena on avata koneoppimisen ja etenkin keinoitekoisin neuroverkoin toteutettavan syväoppimisen käsitteitä, toimintaperiaatetta ja sovelluksia lukijalle, jolla ei ole aiempaa tietämystä alasta.

Koneoppimisen mahdollistaman uuden teknologian käyttökohteet eivät rajoitu pelkästään älypuhelin tai tietotekniikan alalle, vaan koneoppimista hyödynnetään jo nyt useilla eri aloilla lääketieteestä itsejaviin autoihin. Myös teollisuudessa ja prosessitekniikassa voidaan hyödyntää koneoppimista. Tulevaisuudessa koneoppimisen merkitys näyttää tällä hetkellä runsaasti kasvavan monilla toisistaan riippumattomilla aloilla ja arkielämässä, joten on hyödyllistä omata edes alkeellinen käsitys siitä, mitä se on.

Jotta lukijalla on helpompaa seurata tekstiä, määritellään nyt löyhästi tärkeimmät käsitteet:

- **Tekoäly:** Keinotekoinen päätöksiä tekevä tietokonesysteemi, joka hyödyntää koneoppimista. (Mueller & Massaron, 2017, s.19)
- **Koneoppiminen:** Algoritmeja hyödyntävä automaattinen datan käsittelytapa, jolla tekoäly voi kehittyä paremmaksi tehtävässään. Algoritmi on systemaattinen sarja matemaattisia toimenpiteitä, joita suoritetaan syötettävälle tiedolle. (Mueller & Massaron, 2017, s.19-23)
- **Syväoppiminen:** Koneoppimisen osa-alue, joka hyödyntää keinotekoisia neuroverkkoja, joissa (useita) piilotettuja kerroksia. (Brownlee, 2019)
- **Keinotekoinen neuroverkko:** Oppimiseen kykenevä, biologisten aivojen inspiroima verkkomainen tiedonkäsittelyrakenne, joka koostuu neuroneista. (Haykin, 2009 s.31-32)

## 2 KONEOPPIMINEN

### 2.1 Johdatus koneoppimiseen

Koneoppimisalgoritmit ovat olleet olemassa jo muutaman vuosikymmenen, mutta vasta viime aikoina koneoppimista on alettu hyödyntämään suuressa mittakaavassa (Theobald, 2017, s.12). Koneoppimisen etuna on sen kyky käsitellä suurta määrää dataa. Nykyaikana tietotekniikan kehityksen ansiosta, miltei kaikesta voidaan kerätä ja tallentaa suuria määriä tietoa. Tämän kaiken informaation käsittely on tullut liian työlääksi, jopa mahdottomaksi tehtäväksi ihmisvoimin. (Theobald, 2017, s.14) Otetaan esimerkiksi sähköpostin roskapostisuodattimet. Millään yhtiöllä ei olisi taloudellisesti mahdollista palkata satojatuhansia ihmisiä lukemaan käyttäjille saapuvia sähköposteja puhumattakaan mahdollisen tietosuojan tuomista rajoitteista. Ei ole myöskään mahdollista luoda työntekijöiden voimin listaa maailman kaikista mahdollisista sähköposteista, jotka pitää suodattaa. Ongelman ratkaisuun tarvitaan koneoppimista, missä tekoäly pystyy oppimaan ja yleistämään piirteitä, joita löytyy roskapostista.

Koneoppimista voitaisiin kuvata vaikkapa rinnastuksella ihmisen tekemään päätökseen matkata tiukassa aikataulussa juna-asemalle. Henkilöllä on tietyn verran aikaa käytettävissä ja hänen pitää päättää ehtiikö hän kävellä vai täytyykö maksaa taksista. Kokemus on opettanut ihmiselle, paljonko hänellä kuluu aikaa kävelemiseen hyvällä säällä. Lisäksi hän on mahdollisesti yrityksen ja erehdyksen kautta oppinut ottamaan erilaisia olosuhteita huomioon, kuten jos ulkona on ruuhka-aika tai erittäin liukasta. Kaiken lisäksi juna voi olla tietyissä olosuhteissa, kuten runsaassa lumisateessa hyvin suurella todennäköisyydellä myöhässä. Näiden tekijöiden pohjalta ihminen tekisi tässä tapauksessa päätöksen kulkutapojen välillä tavoitellen junaan ehtimistä mahdollisimman halvalla. Vaikka ihminen ei ole syntymästään asti tiennyt eri tekijöiden vaikutusta eikä edes kävelemiseen kuluvan ajan keskiarvoa, hän on voinut toistojen kautta muokata näkemyksiään ja oppia, jolloin hänen tekemänsä arviot tarkentuvat.

Koneoppiminen rakentuu myös samantyyppiselle perusidealle, jossa empiirisestä ennestään tutun datan perusteella voidaan tehdä päätöksiä uusiin tilanteisiin. Kone siis oppii tekemään tarkempia ennusteita mallin avulla. (Theobald, 2017, s.6) Koneoppiminen poikkeaa tavallisesta tietokoneohjelmasta siten, että tavallinen kone toteuttaa vain käskyt,

joita sille suoraan syötetään, kun taas koneoppimisen avulla kone pystyy tuottamaan pelkästään sille syötetyn datan perusteella päätöksiä. Koneoppimisen seurauksena on koneita, jotka tietyssä mielessä voivat matkia ihmisen ajattelua. Koneella on tosin se etu ihmiseen verrattuna, että se pystyy käsittelemään paljon suurempaa määrää dataa, eri skenaarioita ja laskea paljon monimutkaisempia laskutoimituksia nopeammin kuin ihminen. Sovelluksina saadaan tekoälyjä, jotka voivat esimerkiksi voittaa ihmiset shakissa tai osaavat valita laulukilpailun voittajan. (Theobald, 2017, s.8)

## 2.2 Koneoppimisen tekniikat

Koneoppimisessa on eri tekniikoita siihen, miten algoritmi syöttödatan käsittelylle tulisi rakentaa. Tekniikat määrittävät miten opetettu tekoäly tekee päätöksen datasta. Ne voidaan jakaa karkeasti viiteen eri luokkaan sen perusteella, mikä idea niiden taustalla on. Lähteessä tekniikoita kutsutaan eriäviksi heimokunniksi. (Mueller & Massaron, 2017, s.29)

1. Symbolinen päättely perustuu logiikkaan, filosofiaan ja käänteiseen päättelyyn.
2. **Ihmisaivojen toimintaa matkivat tekniikat** ovat nykyään kaikista puhutuimpia ja käsittävät keinotekoiset neuroverkot ja syväoppimisen, joista on myöhemmin lisää tarkemmin.
3. Evoluutioon perustuvissa tekniikoissa paremmat yksilöt selviävät todennäköisemmin seuraavaan sukupolveen, jolloin koneoppiminen tapahtuu ikään kuin luonnonvalinnan tavoin.
4. Bayesialainen päättely perustuu todennäköisyyksiin ja tilastotieteeseen.
5. Analogioihin perustuvat tekniikat käyttävät kernel-metodia löytääkseen kuvioita eli säännönmukaisuuksia datasta.

(Mueller & Massaron, 2017, s.30-32)

## 2.3 Koneoppimisen lähestymistavat

Päättelytekniikan lisäksi täytyy päättää, miten koneoppimisessa tekoälyä opetetaan. Eri tapoja sanotaan lähestymistavoiksi (Approaches engl.) Näistä kolme keskeisintä etenkin syväoppimisen kannalta ovat ohjattu oppiminen, ohjaamaton oppiminen ja vahvistusoppiminen. (Haykin, 2009, s.64)



### 2.3.1 Ohjattu oppiminen

Ohjatussa oppimisessa mallille syötetään opetusdataa, johon liittyy tieto siitä, mihin kategoriaan tai luokkaan sen pitää kuulua. Seuraavaksi tekoäly harjoittelee opetusdatalla ja pyrkii muokkaamaan algoritmiaan siten, että tulokset datalle kehittyvät kohti annettuja oikeita vastauksia. Mikäli kaikki menee suunnitellusti, tekoäly kykenee yleistämään oppimansa myös uuteen opetusdatan ulkopuoliseen dataan. (Haykin, 2009, s.64-65) Esimerkkinä toimii vaikkapa aiemmin mainittu sähköpostin roskapostin suodatus. Käyttäjät merkkavat saapuneista sähköposteista roskaposteja siirtäessään niitä roskapostikansioon ja toisaalta merkkavat tärkeitä roskaksi luultuja viestejä oikeiksi viesteiksi siirtäessään roskapostiin virheellisesti joutuneita viestejä pois sieltä. Tällöin tekoäly saa paljon opetusdataa siitä, mikä on roskapostia ja mikä ei. Opiteen perusteella tekoäly pystyy tunnistamaan ja suodattamaan ainakin tavallisimmat roskapostit. (Theobald, 2017, s.30)

### 2.3.2 Ohjaamaton oppiminen

Ohjaamattomassa opetuksessa ei ole ulkonaista opettajaa, jolloin syötettävään dataan ei ole merkitty tavoiteltuja vastauksia. Sen sijaan tekoälyyn on rakennettu käyttökohteesta riippumaton tavoite, joka on yleensä pyrkiä luokittelemaan dataa. Syötetystä datasta tekoäly oppii eli sen algoritmi muokkautuu tunnistamaan aineiston tilastollisia säännöllisyyksiä ja opiteen perusteella se pystyy esittämään dataa erilaisissa luokissa automaattisesti. (Haykin, 2009, s.67) Luokkia ei ole siis ennestään syötetty koneelle tai niiden olemassaoloa ei ole tiedetty. Esimerkkinä tällaisesta voitaisiin vaikka kerätä tietoa lukioikäisten nuorien ruumiinmitoista. On todennäköistä, että syntyisi kaksi toisistaan poikkeavaa datarykelmää, koska tytöillä ja pojilla on toisistaan erilaiset keskimääräiset ominaispiirteet huolimatta suurestakin vaihtelusta. Vaikka tekoäly ei tiedä mitä sen löytämät luokat edustavat, se pystyisi silti määrittämään kaksi erillistä luokkaa datasta. (Theobald, 2017, s.30-31)

### 2.3.3 Vahvistusoppiminen

Vahvistusoppimisessa tekoäly on osa ympäristöä. Se tekee päätöksiä, joilla on vaikutusta ympäristöön ja tekoälyllä on jokin asetettu tavoite, jota kohti se pyrkii toiminnallaan. Tärkeää tavoitetta kohti pyrkiessä vahvistusoppimisessa on toimintojen määrän minimoiminen, jotta saavutetaan haluttu. Onnistuminen tehtävässä vahvistaa tekoälylle

sen toimineen oikein, jolloin samanlaista toimintatapaa osataan käyttää uudestaan samoissa oloissa. Vahvistusoppimisen haasteena on kuitenkin, ettei ole opettajaa, joka tietäisi optimaalisimman päätöksien ketjun. Tällöin joudutaan miettimään, mikä on se onnistuminen, josta palaute opitusta annetaan. Tekoälyn pitää pystyä havaitsemaan yksittäisen toiminnon vaikutus sen saamaan palautteeseen mutta palaute saattaa tulla vasta pitkän toimintaketjun lopuksi. (Haykin, 2009, s.67)

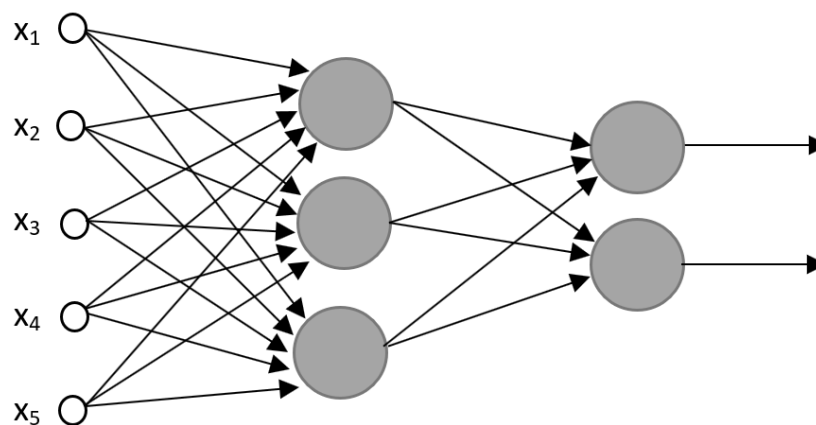
Hyvänä esimerkkinä vahvistusoppimisesta toimii erilaisten pelien pelaamisen opettaminen tekoälylle. Pelejä, joissa palaute tehdystä ratkaisusta tulee välittömästi, voidaan sanoa yksinkertaisemmaksi kuin niitä, joissa ratkaisuilla on kauaskantoisia seurauksia. Yksinkertaisessa pelissä oikea ratkaisu tuottaa heti vaikkapa lisää pisteitä ja väärästä häviää. Tällä tavoin ajateltuna vaikkapa vanhat arcade-pelit, joista esimerkkinä Atari Breakout, ovat yksinkertaisempia kuin vaikka shakki. Yksinkertaisia pelejä on helpompaa opettaa tekoälylle, kuin monimutkaisempia pelejä, koska tekoälyn on helpompi huomata onnistumisensa ja virheensä. Silti molempiin peleihin, sekä Atari Breakout:in (Silver, 2016), että shakkiin (Hassabis et al., 2018) on olemassa tekoälyjä, jotka pystyvät oppimaan pelin pelaamisen mestarilliselle tasolle kehittymällä täysin ensikertalaisesta mestariksi toistojen avulla.

Googlen Deep Mind -ohjelman AlphaZero:n käyttö shakissa on hyvä esimerkki nykyaikaisen koneoppimisen mahdollisuuksista. Aikaisemminkin on ollut shakkitietokoneita, jotka pystyvät kilpailemaan huipputasolla shakissa, kuten IBM:n Deep Blue, mutta aikaisemmat shakkitietokoneet ovat perustuneet tuhansiin tekijöiden käsin koodaamiin sääntöihin ja heuristiikkaan. Eli niiden tekeminen on ollut todella työlästä, eivätkä ne kehity luoviksi, vaan heijastavat tekijäänsä. AlphaZero sen sijaan pohjautuu keinotekoisiiin neuroverkkoihin, jolloin sille ei tarvitse syöttää enempää kuin pelin perussäännöt, jonka jälkeen se oppii paremmaksi harjoittelemalla itseään vastaan. Keinotekoiset neuroverkot voivat olla yleistettäviä eri ongelmien välillä ja ne ovat nykyaikaisen tekoälykehityksen puhutuin aihe. (Hassabis et al., 2018) Seuraavassa kappaleessa käydäänkin läpi peruseriaatteet keinotekoisesta neuroverkosta, sen rakenteesta ja käsitellään lisää sen sovellusmahdollisuuksia.

## 3 KEINOTEKOISET NEUROVERKOT JA SYVÄOPPIMINEN

### 3.1 Mitä on syväoppiminen?

Syväoppiminen on koneoppimisen alalaji, jossa hyödynnetään useita kerroksia sisältäviä keinotekoisia neuroverkkoja. Sana syvä viittaakin juuri neuroverkon useiden kerroksien tuomaan ”syvyyteen” oppimisessa. (Brownlee, 2019) Lyhyesti selitettynä keinotekoinen neuroverkko on rakenne neuroneiksi sanotuista laskentapisteistä. Neuronit on aseteltu eri kerroksiin siten, että ensimmäinen, yleensä vasemmanpuolimmaisiksi piirretty, on syöttökerros ja ulommaisin oikeanpuolimmainen on ulostulosignaalin antava kerros. Välissä on piilotetuiksi kerroksiksi kutsuttuja neuronikerroksia. (Haykin, 2009, s.52) Kuvassa 1 on esitetty yksinkertainen keinotekoinen neuroverkko, mutta sen rakenteen tarkoitus selitetään vasta myöhemmissä kappaleissa auki.



syöttökerros

piilotettu kerros

ulostulokerros

Kuva 1. Monikerroksinen keinotekoinen neuroverkko. Kuvan neuroverkko on täysin yhdistetty 5-3-2-verkko. (mukaillen Haykin, 2009, s.52)

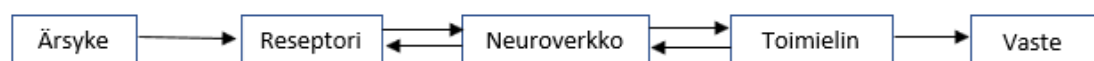
Keinotekoiset neuroverkot ja syväoppiminen ovat saaneet inspiraationsa ihmisaivojen toiminnasta. On huomattu, että aivot eivät toimi perinteisen tietokoneen tavoin, vaan aivot ovat monimutkainen, epälineaarinen ja rinnakkainen tietojenkäsittelyjärjestelmä. (Haykin, 2009, s.31) Aivojen toiminta perustuu hermosoluihin eli neuroneihin ja niiden kykyyn muokata yhteyksiään toisiinsa tarpeen mukaan. Ihmisaivot mahdollistavat muun muassa näkökyvyn. Silmiltä tuleva ärsyke tulkitaan aivoissa ja ihminen pystyy ymmärtämään

katsomaansa. Juuri esimerkiksi kuvantunnistusta pystytään tekemään nykyään tietokoneilla syväoppimisen avulla. Voidaankin sanoa, että keinotekoisella neuroverkolla pyritään mallintamaan aivojen tapaa ratkaista haluttu ongelma. (Haykin, 2009, s.32)

Simon Haykin antaa keinotekoiselle neuroverkolle vapaasti käännettynä määritelmäksi: ”Neuroverkko on suuresti rinnakkaisesti hajautettu prosessori, joka muodostuu yksinkertaisista prosessointiyksiköistä, joilla on luontainen taipumus tallentaa koettua tietoa ja mahdollistaa sen käyttöä.” (Haykin, 2009, s.32) Neuroverkossa, kuten aivoissa opittu tieto tallentuu juuri neuronien välisien yhteyksien vahvuuteen, joita kutsutaan synaptisiksi painoiksi, tosin erikoisemmissä tapauksissa myös uusia yhteyksiä voi syntyä ja vanhoja kadota, jolloin verkon rakenne muuttuu. Käytäntöä, jolla oppiminen keinotekoisella neuroverkolla tapahtuu, kutsutaan oppimisalgoritmiksi, jossa verkon parametrejä muokataan kohti halutun tehtävän suorittamista paremmin. (Haykin, 2009, s.32)

### 3.2 Biologisten aivojen toimintaperiaate

Ihmisen hermosto voidaan jakaa kolmenlaisiin tehtäviin karkeasti. Keskellä on aivot, joita keinotekoiset neuroverkot pyrkivät matkimaan. Aivoissa syntyy päätös toiminnasta. Aivot saavat tietoa ympäristöstä tai muualta kehosta reseptoreista, eli esimerkiksi aisteista. Reseptorit voivat olla teknologiassa vaikkapa mittalaitteita. Aivoista lähtevät signaalit etenevät erilaisille toimielimille, kuten lihaksille, jotka saavat aikaan halutun tehtävän. Toimielimet voivat olla teknologiassa toimilaitteita, kuten venttiili. Signaalit etenevät hermostossa lyhytaikaisina jänniteimpulsseina. Lisäksi systeemissä on takaisinkytkentää neuroverkosta reseptoreille ja toimielimistä neuroverkkoon. (Haykin, 2009, s.36) Ihmisen hermoston perusidea on hahmoteltuna kuvassa 2.



Kuva 2. Hermoston perusidea. (mukaillen Haykin, 2009, s.36)

Itse aivot muodostuvat ihmisellä hermosoluista ja niiden välisistä yhteyksistä. Hermosoluilla on kahdenlaisia haaroja: aksoneita ja dendriittejä. Solut voisi rinnastaa puuhun, johon tulee monihaarainen juuristo ja josta lähtevä yksi runko haarautuu oksiin. Signaalia vastaanottavia haaroja kutsutaan dendriiteiksi (juuret puuanalogiassa) ja

hermosolusta lähtevää signaalia johtavaa haaraa kutsutaan aksoniksi (runko). Edellisen hermosolun aksoni haarautuu ja yhtyy seuraavien solujen dendriitteihin kohdissa, joita kutsutaan **synapseiksi**. Yleisimmässä synapsissa sähköinen signaali muuttuu kemialliseksi rajapinnan ajaksi mutta palaa sitten dendriitissä jälleen sähköiseksi. Kyseisessä asiassa on mielenkiintoista ja keinotekoisille neuroverkoille tärkeää, että kemiallinen viestiaine voi joko kiihdyttää tai inhiboida seuraavaa hermosolua. Tämä näkyy myös keinotekoisessa neuroverkossa, koska edellisen neuronin signaali voi vaikuttaa seuraavaan tuottamaan signaaliin hillitsemällä tai kiihdyttämällä sitä. (Haykin, 2009, s.36-37)

Ympäristöön sopeutuminen eli oppiminen tapahtuu aivoissa **plastisuuden** eli muovautuvuuden avulla. Plastisuus aikuisen aivoissa ei tarkoita uusien hermosolujen syntyä, vaan uusien synapsisten yhteyksien syntyä tai olemassa olevien muokkautumista. (Haykin, 2009, s.36) Ihmisellä paljon käytetyt yhteydet vahvistuvat ja käyttämättömät rapistuvat. Keinotekoisella neuroverkolla usein yhteyksiä muokataan minimoimaan virhettä tai tehdyn työn määrää, jotta asetettu tavoite onnistuu. Molemmista siis tarvitaan plastisuutta oppimiseen. (Haykin, 2009, s.75)

Aivojen rakennetta voidaan tarkastella eri monimutkaisuuden tasoilla. Alimmalla tasolla on vain molekyylijä ja seuraavalla synapseja. Synapsien pienintä järkevän funktion aikaansaavaa yksikköä kutsutaan neuromikropiiriksi. Tätä voidaan verrata transistoreista koostuvaan piimikropiiriin. Neuromikropiirit muodostavat dendriitin osa-alueita, dendriittisiä alayksiköitä, joista koko dendriittijuuristo muodostuu. Koko hermosolu eli neuroni on noin 100 µm kokoinen ja siinä on useita dendriittisiä alayksiköitä. Lähekkäiset neuronit muodostavat paikallisia piirejä (n. 1 mm). Nämä piirit kykenevät suorittamaan tietylle aivon alueelle ominaisen tehtävän. Eri paikalliset piirit pystyvät viestimään keskenään alueiden välisten yhteyksien kautta ja jälleen muodostuu suurempia kokonaisuuksia, kuten näköaistimusta käsittelevä alue, kunnes lopulta puhutaan koko keskushermostosta. Digitaaliset neuroverkot eivät vielä yllä lähellekään aivojen tasolle eri kerrosten monimutkaisuudessa, määrässä ja neuronin hienostuneisuudessa mutta keinotekoisissa neuroverkossakin on erilaisia hierarkisia tasoja. Voidaan ajatella, että keinotekoisilla neuroverkoilla on vielä paljon kehittymisvaraa asian suhteen. (Haykin, 2009, s.37-39)

### 3.3 Neuron

Keinotekoinen neuroverkko koostuu verkosta toisiinsa yhteyksissä olevista digitaalisista neuroneista. Seuraavaksi käydään läpi hieman mallia neuronille ja siinä esiintyvää indeksointia. Neuron on informaation prosessointiyksikkö, eräänlainen laskentapiste, joka koostuu kolmesta osasta. (Haykin, 2009, s.40)

1. Synapsit eli yhteydet, josta tulosignaali  $x_j$  saapuu. Jokaiselle yhteydelle on oma painotuskerroin  $w_{kj}$  (weight engl.), joka saa joko negatiivisen tai positiivisen reaalilukuarvon. Indeksillä  $k$  viitataan tarkasteltavaan neuroniin (neuron  $k$ ) ja  $j$  viittaa monesko yhteys on kyseessä (synapsi  $j$ ). (Haykin, 2009, s.40)
2. Summapiste kokoaa kaikki neuronin synapsien signaalit yhteen ja summaa ne. Summaan voidaan ja yleensä lisätään tulosignaaleista riippumaton vakiotermi  $b_k$  (bias engl.). Tällöin seuraavaan vaiheeseen siirtyvä arvo on  $v_k =$  painotetut tulosignaalit  $+ b_k$ . Arvoa  $v_k$  kutsutaan indusoiduksi paikalliseksi kentäksi (induced local field engl.) (Haykin, 2009, s.40-42)

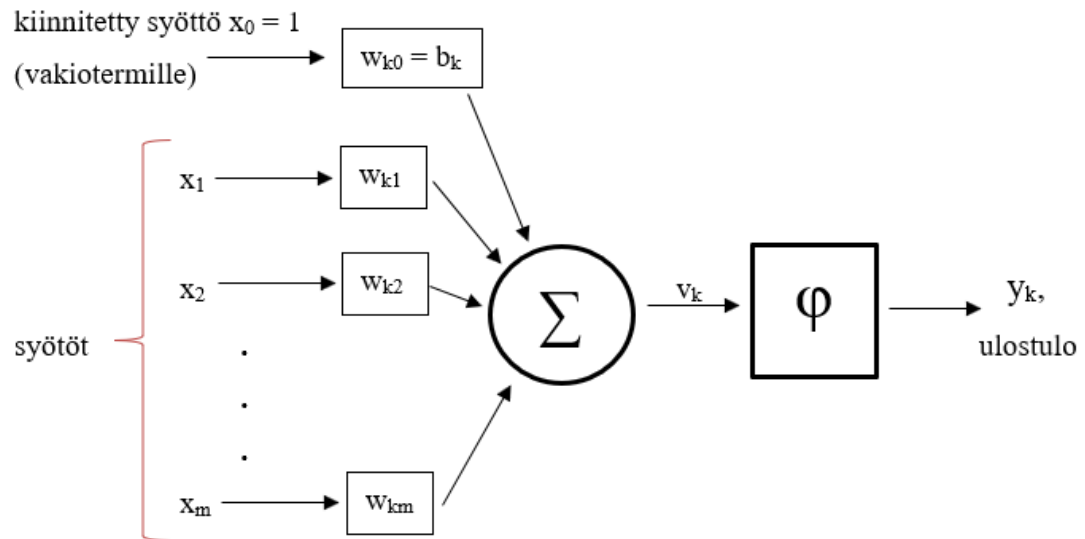
$$v_k = \sum_{j=0}^m w_{kj} x_j \quad (1)$$

Kaava (1) toimii siten, että  $x_0$  on aina 1 ja  $w_{k0} = b_0$ . Muuttujat  $x_1-x_m$  ovat varsinaiset tulosignaalit. Kuva 3 havainnollistaa neuronin toimintaa tarkemmin. (Haykin, 2009, s.41-42)

3. Aktivaatiofunktio  $\varphi$  suorittaa summatulle arvolle matemaattisen operaation, jotta neuronin ulostulo saa arvon sovitun amplitudirajan sisältä, joka on yleensä välillä  $[0,1]$  tai  $[-1,1]$ . (Haykin, 2009, s.40-41)

Neuronin ulostuloa merkitään  $y_k$  ja sille voidaan kirjoittaa seuraava kaava. (Haykin, 2009, s.41)

$$y_k = \varphi (v_k) \quad (2)$$



Kuva 3. Malli neuronille. (mukaillen Haykin, 2009, s.42)

### 3.3.1 Aktivaatiofunktio

Aktivaatiofunktiot voidaan jakaa kahteen perustyyppiin. Raja-arvofunktioihin ja sigmoid-funktioihin. (Haykin, 2009, s.43-44)

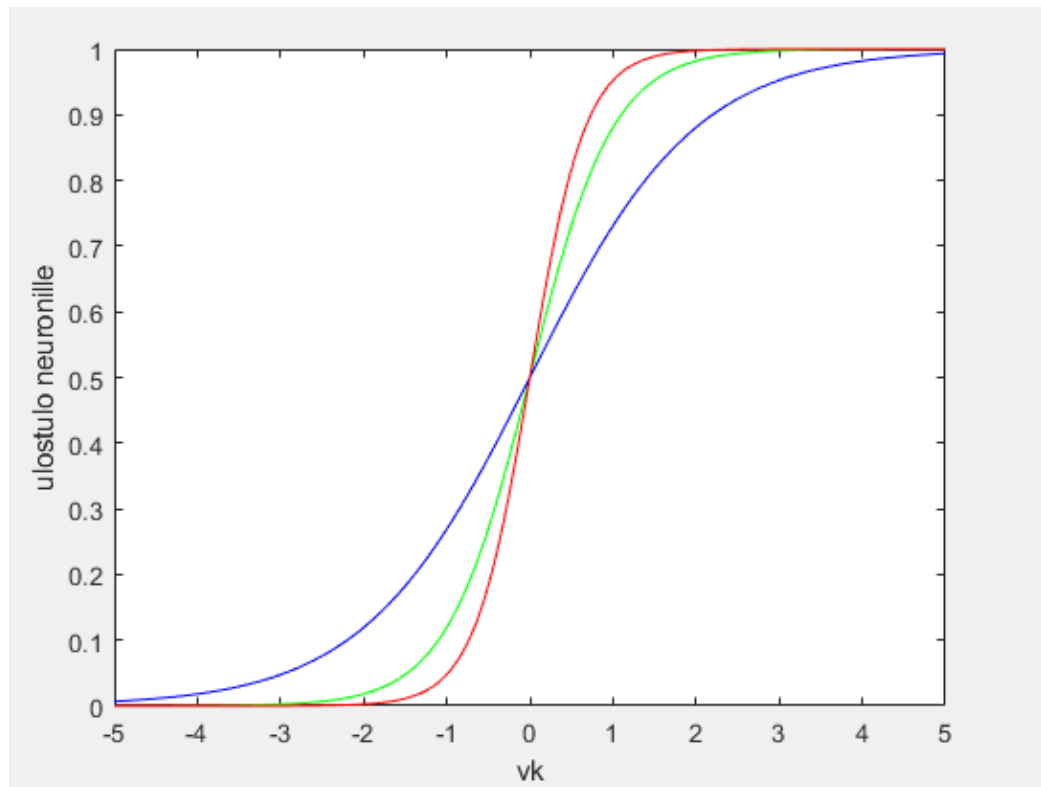
1. Raja-arvofunktiota kutsutaan myös Heavisiden funktioksi ja se saa joko arvon 1 tai 0 riippuen, onko tietty raja-arvo (yleensä nolla) saavutettu tai ylitetty. Toinen vaihtoehto on saada arvo -1, 0 tai 1, jota esitetään kaavalla: (Haykin, 2009, s.43)

$$y_k = \varphi(v_k) = \begin{cases} 1 & \text{jos } v_k > 0 \\ 0 & \text{jos } v_k = 0 \\ -1 & \text{jos } v_k < 0 \end{cases} \quad (3)$$

2. Sigmoid-funktiossa aktivaatiofunktio lähestyy amplitudin maksimia tai minimiä, kun  $v_k$  etenee kauemmaksi nolasta. Erittäin suurilla tai pienillä syöttöarvoilla sigmoid-funktio käyttäytyy käytännössä, kuten raja-arvofunktio, mutta sigmoid-funktio eroaa siinä, että sillä on nollian ympärillä jatkuva, loiva osuus, jolla siirrytään portaattomasti arvosta 0 (joskus -1) arvoon 1. Sigmoid-funktion kuvaaja on S:n muotoinen ja se on käytetyin aktivaatiofunktioityyppi. Ohessa esimerkki sigmoid-funktiosta, joka saa arvoja avoimelta väliltä:  $]0,1[$  (Haykin, 2009, s.44)

$$y_k = \varphi(v_k) = \frac{1}{1 + \exp(-av_k)} \quad (4)$$

Yhtälössä  $a$  on parametri, jolla vaikutetaan siirtymän jyrkkyyteen, kun siirrytään arvosta 0 arvoon 1. (Haykin, 2009, s.43) Parametrin vaikutus ja funktion muoto on esitettyä kuvassa 4.



Kuva 4. Sigmoid-funktioita yhtälön 4 mukaisesti. Punaisella  $a=3$ , vihreällä  $a=2$  ja sinisellä  $a=1$  (punainen jyrkin, sininen loivin).

On myös sigmoid-funktioita, jotka saavat arvoja väliltä:]  $-1,1$ [. Sigmoid-funktion etuna on jatkuvuus ja derivoituvuus, ja erityisesti derivoituvuudesta voi olla suurta etua. (Haykin, 2009, s.43)

### 3.3.2 Stokastiset mallit

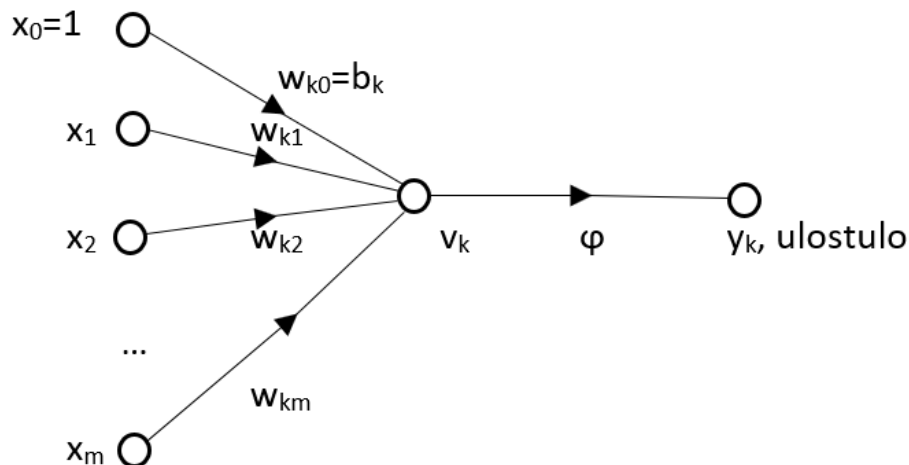
Aktivaatiofunktioiden yhteydessä on myös hyvä mainita, että determinististen eli aina samoilla tuloarvoilla samat lopputulokset antavien mallien lisäksi voidaan neuronin ulostulolle käyttää stokastista eli todennäköisyyteen perustuvaa mallia. Tällöin aktivaatiofunktio edustaa todennäköisyyttä  $P(v_k)$ , jolla ulostulo  $y_k = 1$ . Tällöin aktivaatiofunktio on yleensä sigmoid-funktio, jossa esiintyy parametri, jolla kuvataan systeemissä esiintyvää kohinaa. (Haykin, 2009, s.44-45)



### 3.4 Neuroverkon kuvaaminen kaaviolla

Keinotekoisia neuroverkkoja voidaan kuvata erilaisilla kaavioilla. **Lohkokaaviolla**, signaali-virtauskaaviolla ja arkkitehtuurisella kaaviolla. (Haykin, 2009, s.48) Mikäli piirretään kaikki laskutoimitukset näkyville, kutsutaan kaaviota lohkokaaavioksi. (Haykin, 2009, s.45) Neuronin lohkokaavio on esitettyä kuvassa 3.

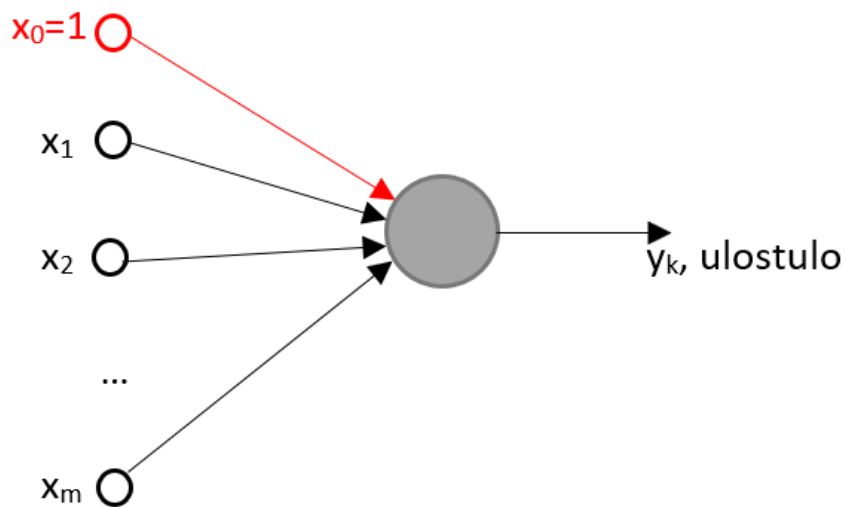
Mikäli taas noudatetaan yksinkertaistavia piirtämissääntöjä, voidaan lohkokaaviosta muodostaa **signaali-virtauskaavio**. Tällöin kaavio koostuu solmukohdista ja niitä yhdistävistä yksisuuntaisista signaalien virtausta kuvaavista nuolista. Nuolen kohdalla signaaliin voi vaikuttaa painotuskerroin tai aktivaatiofunktio. Solmukohtaan tulevat signaalit summataan ja solmukohta lähettää oman arvonsa mukaisen yhtä suuren signaalin jokaiseen lähtönuoleen. (Haykin, 2009, s.45-47) Kuten lohkokaavio, signaalivirtauskaavio antaa täyden tiedon kyseisestä verkosta ja yhden neuronin esittämiseen tarvitaan enemmän kuin yksi solmukohta. Neuronin signaali-virtauskaavio on esitetty kuvassa 5.



Kuva 5. Signaali-virtauskaavio (mukaiillen Haykin, 2009, s.47)

Mikäli halutaan yksinkertaistaa kaaviota pidemmälle, voidaan signaalivirtauskaavio muokata **arkkitehtuurisen kaavion** muotoon. Tällöin koko neuroni yksinkertaistetaan yhdeksi laskentapisteeksi, jonka sisään kuuluvat painotukset, summaaminen, vakiotermi ja aktivaatiofunktio. Nuolet eri neuronien välillä kuljettavat signaalin yksisuuntaisesti sellaisenaan. Ensimmäinen kerros saa signaalinsa tulosolmuista. Tällaisella kaaviolla voidaan kuvata helpommin verkon neuronien välisiä yhteyksiä, ja esitystapaa

käytetäänkin, kun käsitellään kokonaista neuroverkkoa. (Haykin, 2009, s.47)  
Arkkitehtuurinen kaavio on esitetty kuvassa 6.



Kuva 6. Arkkitehtuurinen kaavio. Vakiotermin vaikutusta ei yleensä koko verkkoa tarkasteltaessa piirretä näkyviin, joten se ( $x_0$ ) on merkitty punaisella. (mukaillen Haykin, 2009, s.47)

### 3.5 Verkon rakenne

Keinotekoisien neuroverkon rakenne määrittelee pitkälti sen, miten verkkoa voidaan opettaa, miten se käsittelee tietoa ja siten myös sen mahdollisen käyttökohteen. (Haykin, 2009, s.51) Työssä tarkastellaan kahta erilaista luokkaa verkoille: monikerroksiset eteenpäin kytketyt verkot ja recurrent-verkot.

#### 3.5.1 Monikerroksiset eteenpäin kytketyt verkot

Monikerroksisissa eteenpäin kytketyissä verkossa eteenpäin kytkeminen tarkoittaa, että signaali etenee verkossa vain yhteen suuntaan eli eteenpäin kerroksien välillä. Monikerroksisuus tarkoittaa, että verkossa on ainakin yksi piilotettu kerros, eli kerros neuroneita syöttökerroksen ja ulostulokerroksen välillä. Piilotetuilla kerroksilla päästään nykyaikaisten keinotekoisien neuroverkkojen olennaisimpaan asiaan. Piilotetuilla kerroksilla saadaan tarkasteltua syöttötietoja korkeammalla tasolla, koska piilotettu kerros käsittelee tietoa hyödyllisellä tavalla ennen ulostuloa. Toisin sanoen piilotetuilla kerroksilla neuroverkko voi tarkastella syötetystä tiedosta muodostuvia **kuvioita** tai **kokonaisuuksia**, mitä voidaan kutsua myös globaaliksi näkökulmaksi. (Haykin, 2009, s.52) Suuremmalla määrällä piilotettuja kerroksia saadaan tutkisteltua suuremman

mittaluokan kokonaisuuksia datasta, eli vaikkapa monimutkaisempia kuvioita, joita muodostuu mahdollisesti useista pienistä kuvioista. Eräs esimerkki monikerroksisesta eteenpäin syöttävästä verkosta on esitetty kuvassa 1 sivulla 12.

Yksinkertainen käytäntöön liittyvä esimerkki saadaan jälleen kuvantunnistuksesta, jossa neuroverkon tehtävänä olisi tunnistaa ja luokitella käsinkirjoitettuja numeroita välillä 0-9. Numerot esitetään 28x28-kokoisina mustavalkokuvina. Tällöin **syöttökerrokseksi** on luonnollisinta laittaa  $784 = 28 \times 28$  syöttösolmuketta, jokainen yhdelle pikselille. Yksittäisen syöttösolmukkeen (neuronin lähteessä, mutta Haykin kutsuu ensimmäistä kerrosta solmukkeiksi) saama arvo on sille ominaisen pikselin kirkkaus välillä  $[0,1]$ . (Nielsen, 2015, luku 1)

**Ulostulokerros** kyseiselle verkolle koostuu kymmenestä neuronista. Jokainen niistä edustaa yhtä kokonaislukua nolasta yhdeksään ja niiden saama arvo edustaa verkon varmuutta kyseiseen numeroon. Arvo 1 ulostulokerroksen neuronille tarkoittaa, että verkko on täysin varma, että kuva on luku, jota neuroni edustaa. Arvo nolla taas tarkoittaa, että kuva ei edusta kyseistä lukua verkon tulkinnan perusteella. Neuroverkon tekemä lopullinen valinta on neuroni ulostulokerroksessa, joka saa suurimman arvon. (Nielsen, 2015, luku 1)

Välissä on tässä tapauksessa yksi **piilotettu kerros**, joka koostuu 15 neuronista, mutta neuronien määrä ei määrity mistään säännöistä ja eri määrätkin voisivat toimia. Piilotetun kerroksen neuronien tehtävänä on tunnistaa syötteestä erilaisia kuvioita, ja teoriassa ensimmäinen 15 neuronista voisi vaikkapa syttyä, kun syötteessä on luvun 9 yläosassa esiintyvä rinkula. Toinen neuroni voisi aktivoitua vaikkapa, kun syötteessä on luvun 9 rinkulasta lähtevä pitkä suora. Tällöin kun kaksi ensimmäistä neuronia aktivoituu, pitäisi niiden perusteella aktivoitua ulostulokerroksen neuroni, joka vastaa lukua 9. Jokaisella piilotetun kerroksen neuronilla on oma kuvionsa, joka aktivoi sitä. (mukaiillen Nielsen, 2015, luku 1)

Piilotetun kerroksen neuronien tarkkailemat kuviot perustuvat niiden yhteyksien vahvuuteen yksittäisiin edellisen kerroksen neuroneihin eli painotuksiin. Esimerkiksi jossain toisessa sovelluksessa, yksi piilotetun kerroksen neuroni voi painottaa oikean yläkulman pikseleitä vastaavien syöttösolmukkeiden signaalit tärkeiksi. Painotukset eivät ole ennalta suunnittelijan rakentamia, vaan ne määrittyvät verkon oppimisen aikana.

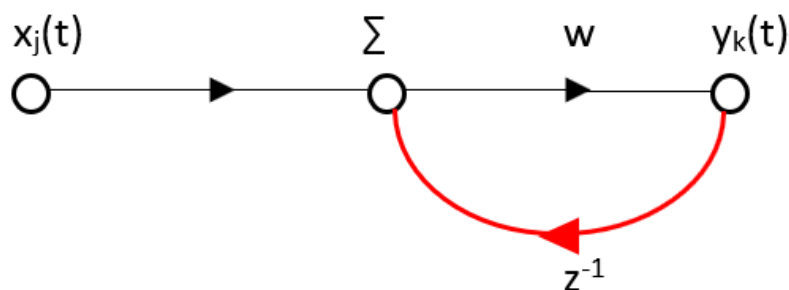
Tällöin viimeksi kuvattu tapa luvun 9 kuvaamiseen on hyvin keinotekoinen, sillä neuroverkkoon voi löytyä opetuksen aikana hyvin erilaisia piirteitä tärkeiksi, kuin mitä ihminen käyttäisi lukujen kuvaamiseen.

Monikerroksisia eteenpäin syöttäviä verkkoja on **täysin yhdistettyjä**, jolloin jokainen edeltävän kerroksen neuroni tai solmuke yhdistyy jokaisen kerroksen neuroniin. Toinen vaihtoehto on **osittain yhdistetyt verkot**, jolloin osa yhteyksistä puuttuu. Verkkoja voidaan nimetä niiden kerrosten sisältämien solmukkeiden ja neuroneiden määrän mukaan. Edellisen esimerkin kuvantunnistusverkkoa kutsuttaisiin 784-15-10 verkoksi, josta käy ilmi kerrosten määrä, sekä niissä olevat syöttösolmukkeet ja neuronit. (Haykin, 2009, s.52-53)

### 3.5.2 Recurrent-verkot

Recurrent-verkko eroaa eteenpäin syöttävästä verkosta siinä, että siinä on vähintään yksi **takaisinkytketty piiri**. Verkossa voi olla useampia kerroksia neuroneita, joten recurrent-verkko voi olla myös monikerroksinen. Takaisinkytkentä tarkoittaa, että neuronin oma ulostuloa syötetään takaisin sen sisääntulosignaalksi, millä on suuri vaikutus verkon ominaisuuksiin. Tällaisessa verkossa joudutaan tarkastelemaan takaisinkytketyn systeemin dynamiikkaa, jota monimutkaistaa usein epälineaarinen systeemi. (Haykin, 2009, s.53)

Tarkastellaan yhtä suljettua takaisinkytkettyä piiriä kahden neuronin välille signaali-virtauskaaviolla kuvassa 7.



Kuva 7. Takaisinkytketty piiri kahden neuronin välillä. (mukaillen Haykin, 2009, s.49)

Tässä tapauksessa  $y$ :n ulostuloon vaikuttaa  $x$ :n lähettämä signaali, johon on summattu takaisinkytkentä  $y$ :n ulostulosta yksi aikavyöhyke sitten. Muuttuja  $z^{-1}$  kuvaa aikaviivettä, joka syntyy tiedon ulostulosta lähettämisestä takaisin. Summa kerrotaan

painotuskertoimella  $w$ :llä. Olennaista kyseisessä tapauksessa on systeemin **stabiilisuuden tarkastelu** painotuskertoimen kannalta. Stabiilisuudella tarkoitetaan, että vaimeneeko systeemiin kerran tuotu signaali vai vahvistuuko se hallitsemattomasti signaalin tuomisen lakattua. Haykinin kirjassa johdettujen kaavojen perusteella saadaan, että systeemi on stabiili, jos painotuskertoimen itseisarvo on alle 1 ja epästabiili jos se on yli 1. Mikäli  $w$ :n itseisarvo on 1, systeemi on stabiilisuusrajalalla. (Haykin, 2009, s.48-49)

### 3.6 Tiedon esittäminen verkolle

Keinotekoisien neuroverkon käyttökohteesta riippumatta verkon on pystyttävä muodostamaan toimintaympäristöstään jonkinlainen matemaattinen kuvaus eli malli. Mallin avulla tietyt sisääntulot johtavat haluttuihin ulostuloihin. Malli rakentuu verkon rakenteisiin ja sen muodostamiseen tarvitaan tietoa toimintaympäristöstä ja tätä **tietoa on kahdenlaista**. Ensimmäinen on etukäteen verkon suunnittelijoiden määrittelemää tietoa, joka on verkkoon rakenteeseen valmiiksi asetettuja arvoja ja rakenteita. Toinen on tietoa, jota saadaan mittauksien ja opetuskierroksien kautta ympäristöstä. (Haykin, 2009, s.54-55)

**Mittauksista saatava tieto** sisältää niin kutsuttuja esimerkkejä yksittäisistä mitatuista tapahtumista. Esimerkit voivat olla merkittyjä tai merkitsemättömiä. Tällä tarkoitetaan sitä, että esimerkkien mukana voi tulla tai olla tulematta samalla tieto tavoitellusta verkon vasteesta. Merkityissä esimerkeissä verkko saa samalla tiedon, mikä niille haluttu tulos on ja merkitsemättömille taas ei. Merkittyä dataa voidaan käyttää neuroverkon ohjattuun oppimiseen opetusdatana. (Haykin, 2009, s.55) Jos kaikki saatavilla olevasta datasta on merkitsemätöntä, eli ei tiedetä etukäteen haluttua vastetta, joudutaan soveltamaan jotain muuta lähestymistapaa kuin ohjattua oppimista. Vaikkapa viime kappaleen piirrettyjä numeroita tulkitseva neuroverkko tarvitsisi hyvin paljon merkittyä dataa oppiakseen hyvän mallin. Joissakin keinotekoisien neuroverkon sovelluksissa, joissa käytetään ohjattua oppimista voikin olla haasteena saada tarpeeksi merkittyä opetusdataa.

Tärkeä ero keinotekoisien neuroverkkojen ja muiden ”perinteisempien” koneoppimisen haarojen välillä onkin juuri syötettävässä tiedossa, kuten sen luonteessa ja määrässä. **Perinteisemmässä koneoppimisessa** tekoälylle täytyy syöttää paljon enemmän säännönmukaisuuksia ja muuta etukäteistä tietoa haluttavasta mallista, että päästään alkuun. Tämä on ollut vaikeaa, koska esimerkiksi tarkkojen mallien kehittäminen käsin

piirretyille numeroille on erittäin haastavaa ja työlästä. Toisaalta perinteisemmässä koneoppimisessa opetusdataa tarvitaan huomattavasti vähemmän, koska tekoälyllä on jo valmiiksi pidemmälle muodostettu malli ympäristöstä ennen ensimmäisen esimerkin syöttämistä sille. (Haykin, 2009, s.55)

Syväoppimisessa keinotekoisien verkkojen rakenteen edustama malli ympäristöstä muokkautuu kohti tarkkaa mallia **opetusdatan avulla** hyvin huonosta ja epätarkasta lähtötilanteesta. Tämä on suuri etu, koska tällöin verkkojen suunnittelijoilla ei tarvitse olla tiedossa tarkkaa matemaattista kuvausta halutusta kohteesta valmiina vaan neuroverkko ratkaisee sen itselleen muokkaamalla verkkojen vapaita parametreja, eli synaptisia painotuksia ja vakiotermejä. Keinotekoisia neuroverkkoja voidaan siis käsitellä black box-mallina ainakin sen onnistuessa. Huonona puolena on, että keinotekoinen neuroverkko tarvitsee valtavan määrän opetusdataa ja opetusdata täytyy kuvata monipuolisesti ympäristöä, johon verkko on tarkoitettu. (Haykin, 2009, s.55-56) Esimerkiksi tehtaalla laadunvalvontaan oppivan neuroverkon pitää saada paljon opetusdataa, jopa hyvin harvinaisista vikatuotteista, eikä vain suurta määrää tavallista dataa.

Keinotekoisien neuroverkkojenkin on kaikesta huolimatta tärkeä pystyä rakentamaan **valmiiksi annettua tietoa**. Verkkojen rakennetta voidaan ohjata tiettyyn suuntaan kahdella tavalla. Yhteyksiä neuroniin tulevia signaaleita varten eli sen vastaanottokenttää voidaan rajoittaa, jolloin päätetään, että tietyt alueet syötöstä ovat epäolennaisia tietyille neuroneille poistamalla ne. Täysin yhdistetyistä verkoista tehdään siis osittain yhdistettyjä. Toinen vaihtoehto on rajoittaa painotuskertoimien valintaa jakamalla samat painotuskertoimien arvot usean neuronin välillä. Tällaista kutsutaan konvoluutioneuroverkoksi (convolutional neural network engl.). Molemmilla keinoilla verkkoon saadaan vähemmän vapaita parametreja, joten verkko oppii nopeammin, sen käyttö on laskennallisesti kevyempää ja sen kehittäminen on helpompaa. (Haykin, 2009, s.58-60)

Keinotekoisien neuroverkkojen rakenteen optimointiin ei ole ainakaan toistaiseksi olemassa teoriaa. Tämän takia keinotekoisien neuroverkkojen suunnittelun voi väittää olevan myös osittain taidetta tieteen lisäksi. Haykinin mukaan toimivin ratkaisu löytyy usein erilaisten kokeilujen kautta ja muilta saadun kokemuksen avulla. (Haykin, 2009, s.64)

## 3.7 Oppiminen

Keinotekoinen neuroverkko oppii muokkaamalla verkossa esiintyviä vapaita parametreja. Minkä perusteella vapaita parametreja muokataan, määrittyy oppimisen lähestymistavasta, jota käytetään kyseisessä sovelluksessa. Eri lähestymistavat esiteltiin lyhyesti koneoppimisen yhteydessä, joten nyt käsitellään tarkemmin erästä oppimisalgoritmia ohjatun oppimisen yhteydessä. Täten voidaan myös selittää, ettei tekoäly ole tietoinen vaan pelkästään monimutkainen laskukone. Lisäksi käydään läpi keinotekoisien neuroverkkojen oppimiseen liittyvä ongelma, jota sanotaan ylioppimiseksi.

### 3.7.1 Backpropagation-algoritmi

Tarkastellaan ohjattua oppimista backpropagationiksi kutsutulla oppimisalgoritmilla aiemmin mainitun piirretyn numeron tunnistamisen yhteydessä. Tällaisen yksinkertaisen keinotekoisien neuroverkon vapait parametrit olisivat oppimisen alussa täysin satunnaisia, joten ulostulokerroksen tuloksetkin olisivat. Koska oikeat vastaukset eli halutut ulostulokuviot esimerkeille tiedetään, voidaan jokaiselle keinotekoisien neuroverkon antamalle vastaukselle eli ulostulokuvioille määrittää virhearvo. Opetusdataa syötetään pienissä erissä laskemisen helpottamiseksi ja jokaisen erän jälkeen lasketaan keskiarvo virheelle, ja sen perusteella vapait parametrit muokataan pienentämään virhettä. (Nielsen, 2015, luku 1)

**Virhefunktio** on vapaista parametreista riippuva funktio ja sen saamat arvot lasketaan tässä tapauksessa keskiarvotetulla neliövirheen menetelmällä (mean square error engl.). Jokaiselle ulostuloneuronille (eli neuronille, jotka esittävät verkon vastauksen varmuutta jostain luvusta 0-9) on jokaisessa esimerkissä oma verkon antama ulostulo ja haluttu ulostulo. Menetelmässä virhe lasketaan korottamalla erotus toiseen, minkä jälkeen eri neuronien arvot summataan. Tämän lisäksi menetelmässä pääpiirteittäin otetaan keskiarvo opetuserän kaikkien esimerkkien väliltä. (Nielsen, 2015, luku 1)

$$C(w^*, b^*) = \frac{1}{2n} \sum_{x^*} \|y^*(x^*) - a^*\|^2, \quad (5)$$

missä  $C$  on virhefunktio

$w^*$  vektori, joka edustaa kaikkia verkon painotuksia (weights engl.)

$b^*$  vektori, joka edustaa kaikkia verkon vakiotermejä (biases engl.)

$n$  on opetuserän esimerkkien määrä

$x^*$  on yksittäisen esimerkin tulosignaalin vektori

$y^*$  on vektori halutuista viimeisen kerroksen neuroneiden ulostuloista

$a^*$  on vektori näiden neuroneiden oikeasti toteutuneista ulostuloista

(Nielsen, 2015, luku 1)

Seuraava askel on määrittää **gradientti** eli osittaisderivaatta jokaisen parametrin suhteen virheelle. Gradientti ilmoittaa jyrkimmän nousun suunnan virheelle parametrien suhteen, joten negatiivisen gradientin suunnalla ja verkon suunnittelijan päättämällä askelpituudella päästään kulkemaan kohti tarkempia vastauksia eli pienempää virhettä, toivottavasti kohti globaalia eikä vain lokaalia minimiä. (Nielsen, 2015, luku 1) Virhefunktioita voidaan verrata mäkiseen maastoon, jonka pohjalle halutaan päästä. Gradientti taas ilmaisee mäen nousun tai laskun eri suuntiin tarkasteltavassa kohdassa.

Gradienttia ei tosin pystytä määrittämään yhden pisteen avulla, joka tällä hetkellä on olemassa opetuserän jälkeen. virheen suuruus ja käytössä olevat parametrit Tiedetään mutta ei virhettä pisteen ympärillä. Tähän ongelmaan otetaan avuksi menetelmä, jota kutsutaan **backpropagation-algoritmiksi**. Menetelmässä otetaan huomioon kaikkien ulostuloneuronien virheet tarkastelemalla jokaisen yksittäisen ulostulokerroksen neuronin virhettä ja lähdetään kulkemaan siihen vaikuttavien tekijöiden reittiä takaisinpäin verkossa. Sanotaan esimerkiksi, että neuroverkkoon syötettiin kuva numerosta 5 ja neuroverkko antaa ulostulon, jossa numeroa 4 vastaava ulostuloneuroni on täysin aktivoitu ja muut täysin sammuneita. Tämän perusteella haluttaisiin muuttaa verkkoa siten, että 5 neuronin aktivoituminen kasvaa jyrkästi ja 4 neuronin pienenee jyrkästi. Muita ei haluttaisi muutettavan. (Nielsen, 2015, luku 2)

Ulostulokerroksen vastausta numerolle 5 tarjoava neuronin aktivaation kasvattaminen koostuu kolmesta tavasta. Ensimmäinen on vakiotermin muuttaminen eli tässä tapauksessa kasvattaminen suoraan halutussa neuronissa. Seuraava on muokata



painotuksia edelliseen kerrokseen siten, että vastauksen 5 neuronin saapuvat yhteydet positiivisista aktivaatioista vahvistuvat ja negatiivisista heikkenevät. Kolmas menetelmä on edellisen kerroksen aktivaatioiden muokkaaminen siten, että neuronit, joiden painotuskerroin on positiivinen vastauksen 5 kanssa muuttuvat positiivisemmiksi ja negatiivisia yhteyksiä pitkin saapuvat aktivaatiot heikkenevät. Edellisen kerroksen aktivaatioihinkaan ei voida vaikuttaa suoraan muuten kuin vakiotermin avulla, joten joudutaan taas tarkastelemaan yksittäinen neuronin kerrallaan haluttuja muutoksia painotuksiin ja niihin vaikuttavia neuroneita taaksepäin, kunnes saavutaan syöttökerrokseen asti. Vastauksen neuronin lisäksi kuljettaisiin kaikkien muidenkin ulostuloneuronien kannalta verkko läpi ja tehtävät muutokset olisivat verrannollisia ulostuloneuronin virheen suuruuteen. (mukaillen Sanderson, 2017)

Yhteenvedon backpropagationissa kuljetaan verkko toiseen suuntaan, ja tehdään muutoksia rakenteen vapaisiin parametreihin sen perusteella, minkälaiseksi ulostulo halutaan muokattavaksi. Muutokset auttavat koko verkkoa parempaan suuntaan kaikille eri kuville, koska muutoksia tehdessä algoritmi ottaa huomioon virheen keskiarvon useista esimerkeistä ja kaikki vaikutusreitit parametreihin.

### 3.7.2 Ylioppiminen

Eräs oppimiseen liittyvä piirre on ylioppimisen ongelma. Ylioppimisessa neuroverkko alkaa mukautua yksittäisiin opetuseriin liian tarkkaan, ja se menettää kykyänsä yleistää. Neuroverkon virhe riippumattomalla datalla siis kasvaa, jonkin pisteen jälkeen jatkettaessa opettamista opetusdatalla. Tämän vuoksi keinotekoisista neuroverkkoista suunniteltaessa on tärkeää verkon rakenteen ja tiedon esittämisen lisäksi tehdä oikeanlaisia päätöksiä opetuserien koosta ja opettamisen lopettamiskriteereistä. (Rosin & Fierens, 1995)

On arvioitu, että olisi olemassa oikeat päätettävät parametrit, joilla ylioppiminen neuroverkossa minimoitaisiin, mutta näiden löytymistä vaikeuttavat monet tekijät. Neuroverkkojen kyky oppia ei näet seuraa aina sen teoreettisesti arvioitua kykyä. Lisäksi ylioppiminen ei ole välttämättä koko verkkoon kerralla vaikuttava ilmiö, vaan tietyt osat verkosta voivat alkaa ylioppimaan, vielä kun toisissa osioissa tarvitaan vielä lisää opettamista. (Rosin & Fierens, 1995)

### 3.8 Käyttökohteita

Keinotekoisilla neuroverkoilla on jo nykyään useita mielenkiintoisia käyttökohteita puhumattakaan siitä mitä tulevaisuudessa on mahdollista tehdä. Seuraavaksi esitellään erilaisia olemassa olevia käyttökohteita keinotekoiselle neuroverkolle ja tarkastellaan tarkemmin niiden käyttämistä teräksenvalmistuksessa. Lopuksi luetellaan joitakin ominaisuuksia, jotka antavat keinotekoisille neuroverkoille etua sovelluksissa.

Haykin esittelee kirjassaan viisi perustyyppiä keinotekoisien neuroverkon käyttökohteille. Ensimmäinen on **kuvion assosioiminen** eli verkon opettaminen yhdistämään tietyt kuvat toisiinsa. Kun verkolle näytetään ”avainkuviot” se muistaa ja tuottaa halutun ulostulokuvion. Toinen käyttökohde on **kuvion tunnistus**, kuten jo moneen kertaan esitelty kuvantunnistus, jossa syöttöjä luokitellaan. (Haykin, 2009, s.68-70)

Kolmas käyttökohde on **toimintaympäristöä kuvaavan funktion approksimointi**. Tällöin on jokin systeemi, vaikkapa teollinen, josta tiedetään syöttötietoja ja ulostuloja muttei ole olemassa mallia systeemille. Malli voidaan approksimoida keinotekoiseen neuroverkkoon ohjatun oppimisen avulla antamalla verkolle syöttötiedot systeemistä ja käyttämällä tiedettyjä ulostuloja opetusdatana. Tästä voi olla suurta etua esimerkiksi teollisten MIMO-prosessien (multiple input-multiple output engl.) mallintamiseen. (Haykin, 2009, s.70-71)

Neljäs käyttökohde on systeemin tai prosessin **säätö**. Tällöin neuroverkon syöte olisi virhesignaali, joka koostuu systeemin halutun ja reaalisen ulostulon erotuksesta. Neuroverkon ulostulona olisi signaalit erilaisille toimilaitteille, joilla vaikutetaan systeemiin. (Haykin, 2009, s.72-73)

Viides käyttökohde on **signaalien kohinan suodattaminen**. Tällöin keinotekoisien neuroverkon tehtävänä on erottaa syötettävästä lähetyksestä halutut taajuudet tai osat ja päästää vain ne läpi. Tällaisia neuroverkkoja voidaan hyödyntää esimerkiksi tutkasysteemeissä. (Haykin, 2009, s.72-73)

Tarkastellaan seuraavaksi keinotekoisien neuroverkkojen käyttöä **teräksenvalmistuksessa** prosessin mallintamisen työkaluna. Eräässä kohteessa tehtävänä on pystyä ennustamaan masuunissa raakaraudan piipitoisuus syöttömuuttujien

avulla. (Saxén & Petterson, 2007; Mäkelä, 2019 s.15 mukaan) Ongelmaan on pystytty hyödyntämään keinotekoista neuroverkkoa, joka oppii karsimaan epärelevantit lähtömuuttujat ja ottaa huomioon prosessin viiveet. Neuroverkon suorittamaa epärelevanttien lähtömuuttujien karsintaa sanotaan karsinta-algoritmiksi ja tuloksena saadaan tietää, mitkä ovat keskeisimmät syöttömuuttujat ja mallintamiseen soveltuvan verkon rakenne. (Saxén & Petterson, 2006; Mäkelä, 2019 s.15 mukaan) Täysin yhdistetystä neuroverkosta siis muokkautuu osittain yhdistetty verkko. Mallin haasteena on lokaaliin optimiin eli lokaaliin virheen minimiin jumittuminen, ja mallin on todettu selittävän 55 % piipitoisuuden muutoksista. Tarkempi tieto piipitoisuudesta mahdollistaa masuunin ohjaamisena optimaalisissa olosuhteissa ja kaksin kulutusta voidaan vähentää. (Saxén & Petterson, 2007; Mäkelä, 2019 s.15-16 mukaan)

Keinotekoisilla neuroverkoilla on paljon ominaisuuksia, jotka tekevät niiden käytöstä houkuttelevia. Aiemmin tiedon esittämisen yhteydessä esitettyjen seikkojen lisäksi muita keskeisiä ominaisuuksia ovat muun muassa:

### 1. Muokkautuvuus

Neuroverkoilla on kyky muokata synaptisia painojaan uudessa ympäristössä, jolloin eri ympäristöön opetettu voi oppia uuden ympäristön toimintamallin ilman uuden neuroverkon kehittämistä. (Haykin, 2009, s.33-34)

### 2. Tieto vastauksen epävarmuudesta

Neuroverkko voi kuvion tunnistuksessa antaa ulostulonsa lisäksi tiedon epävarmuudestaan kyseiseen lopputulemaan, jolloin epämääräiset vastaukset voidaan esimerkiksi hylätä tai tarkastella tarkemmin. (Haykin, 2009, s.34)

### 3. Kontekstin vaikutus lopputulokseen?

Koska neuroverkon eri osat vaikuttavat toisten osien toimintaan, pystyy neuroverkko käsittelemään tietoa, jossa konteksti on tärkeää. (Haykin, 2009, s.34) Esimerkiksi kuvantunnistuksessa tietty kuvio voi viitata johonkin lopputulemaan, mutta muu ympäristö paljastaa, ettei ole kyse siitä.

#### 4. Soveltuvuus suuren mittaluokan ongelmille

Rinnakkaisuus neuroverkossa auttaa tiettytyyppisten laskutoimitusten verrattain nopeassa laskemisessa ja mahdollistaa VLSI-teknologian (very-large-scale-integrated engl.) käytön. (Haykin, 2009, s.34)

#### 5. Universaalius eri neuroverkkojen välillä

Eri neuroverkot rakentuvat kaikille yhteisistä neuroneista, jolloin teorioita ja algoritmeja voidaan jakaa eri verkkojen välillä. Lisäksi eri neuroverkkoja voidaan kytkeä yhteen muodostamaan modulaarisia verkkoja. (Haykin, 2009, s.34)

## 4 POHDINTOJA TULEVAISUUDENKUVISTA

On vääjäämätöntä, että tulevaisuudessa keinotekoiset neuroverkot ja koneoppiminen ylipäänsä tulevat kehittymään suuresti. Kun koneoppimisella keksitään ratkaista uudenlaisia ongelmia ja kehitetään kehittyneempiä tekoälyjä, on sillä myös valtava merkitys ympäröivään yhteiskuntaan. Kuten kaikilla uusilla asioilla, myös tekoälyn kehittämisellä voidaan nähdä sekä hyviä puolia että uusia haasteita.

Tulevaisuudessa tullaan näkemään vielä monimutkaisimpia keinotekoisia neuroverkkoja ja sovelluksia. Näillä pystytään ratkaisemaan ongelmia, joita voidaan nykyään pitää miltei mahdottomina. Suurimääräisesti luonnehdittuna koneoppimisella voidaan muun muassa parantaa terveydenhuoltoa, tietojen keräämistä ja käsittelyä, tieteen tekemistä monella alalla, kuluttajapalveluita, robotiikkaa, teollisia järjestelmiä ja ylipäänsä työn tuottavuutta. Tekoälyllä on ympäristön suojelunkin kannalta hyötyjä, sillä keinotekoisilla neuroverkoilla voidaan esimerkiksi kehittää parempia ilmastomalleja. Tämän lisäksi tekoälyllä voidaan tehostaa olemassa olevien luonnonvarojen käyttöä ja kehittää kiertotaloutta. Työssäkävijöiden näkökulmasta koneoppimisesta on hyötyä siinä suhteessa, että tekoäly voi tehdä ihmisten puolesta erityisen tylsiä ja vaarallisissa olosuhteissa ongelmanratkaisua vaativia töitä, minkä lisäksi voi avautua uusia mahdollisuuksia korkeakoulutetuille koneoppimisen kehittämisessä.

Koneoppiminen ja syväoppimien osana sitä asettaa kuitenkin myös haasteita yhteiskunnille. Kuten teollinen vallankumous 1700- ja 1800-lukujen vaihteesta lähtien, puhutaan nykyään olevan käynnissä digitalisaation ja informaatioteknologian tuoma vallankumous. Tämä tulee voimistumaan entisestään, kun koneoppiminen siirtyy yhä enemmän kehitysasteelta työpaikoille. Vaikka työn tuottavuuden kasvaessa valtion bruttokansantuote kasvaa, lisääntynyt varallisuus ei välttämättä hyödytä kaikkia. Erik Brynjolfsson ja Andrew McAfee kertovat kirjassaan, että vuodesta 1999 lähtien USA:ssa mediaanitulo on itse asiassa tippunut, vaikka maan bruttokansantuote ja työn tuottavuus on kasvanut. Tämä kertoo varallisuuden kasvavasta epätasaisesta jakaantumisesta. USA:ssa esimerkiksi vuosien 1983 ja 2009 välissä inflaatiokorjatut tulot kaikille muille kuin rikkaimmalle 20 %:lle laskivat nettona, kun taas rikkain 20 % kasvatti tulojaan yli 100 %. (Brynjolfsson & McAfee, 2014, luku 9)

Koneoppiminen kykenee varmasti lisäämään tulevaisuudessa tuottavuutta ja vähentämään ihmistyövoiman tarvetta, mikä asettaa uhkia työelämän rakenteisiin. Koneoppimisella on tosin erityinen piirre siinä, että sen avulla automaatio ei veisikään enää vain toistettavuuteen perustuvia teknillisiä työtehtäviä, vaan se voi haastaa työpaikkoja ennen koneille liian vaikeiksi nähdyltä aloilta. Palvelualalla ja korkeakoulutetuissa asemissa on töitä, joita tulevaisuudessa saatetaan tehdä yhä enemmän tekoälyn eikä ihmisen toimesta. Esimerkkeinä voidaan mainita myyjät, kuljettajat, terveydenhuollon ammattilaiset tai jopa välitason johtajat. Tämä luo koneoppimiseen ja keinotekoisiiin neuroverkkoihin liittyvän uhan suuriin rakennemuutoksiin ja varallisuuden epätasaiseen jakautumiseen yhteiskunnissa.

Muita koneoppimisen vaaroja ja ongelmia näkeekin runsaasti esimerkiksi elokuvissa, kuten tekoälyn käyttö **sotilaallisiin** tarkoituksiin, sen **riistäytyminen** käsistä tai tekoälyn **tietoisuuteen** liittyvät eettiset kysymykset. Monet kuuluisat teknologian tai tieteen parissa työskentelevätkin henkilöt ovat ilmaisseet huolensa tekoälyn kehittymiseen liittyvistä vaaroista ihmiskunnalle. Internetissä on muun muassa olemassa vetoamuksia kieltää itsenäisten aseiden käyttö, mistä huolimatta useat maat kehittelevät ja jopa omaavat niitä. (Mueller & Massaron, 2017 s.11)

Tekoälyn riistäytyminen käsistä koskee erityisesti aseita, mutta myös muunlaisia tilanteita, kuten esimerkiksi itseajavien ajoneuvojen luotettavuutta ja turvallisuutta. Koska tekoälyllä ei ole ainakaan vielä tietoisuutta, täytyy suunnittelijan tarkkaan miettiä, minkälaisen tavoitteen koneoppimiselle asettaa ja miten se tulee toimimaan käytännössä, koska muutoin tekoäly saattaa päätyä tavoittelemaan jotain täysin eriä mitä suunnittelija halusi. Kuten jos asettaisimme parturirobotin tekoälylle tavoitteeksi vain hiusten leikkaamisen mahdollisimman lyhyessä ajassa, se ei osaisi ottaa huomioon, että korvia ei saa vahingoittaa samalla.

Tekoälyn tietoisuus on nykyään vielä filosofista pohdintaa ja spekulointia eikä ajankohtainen yhteiskunnallinen kysymys. Kuitenkaan ei ole ainakaan periaatteessa mahdotonta, että tulevaisuudessa voitaisiin kehittää niin monimutkaisia tekoälyjä, että ne haastaisivat ihmisten käsityksen tietoisuudesta. Tällöin jouduttaisiin miettimään, milloin jokin systeemi jäljittelee tietoisuutta ja milloin se olisi saavuttanut sen, jolloin sille kuuluisi mahdollisia oikeuksia. Asian pohdintaa tosin vaikeuttaa, etteivät ihmiset ole yksimielisiä, mikä tietoisuuden aiheuttaa biologisilla eliöilläkään, ja tekoälyn

kehittyminen tietoiseksi, joko näennäisesti tai todellisesti, voisi haastaa yhteiskuntarauhaa kyseenalaistamalla muun muassa uskontojen tarjoamia näkemyksiä tietoisuudesta.

Kaiken kaikkiaan tarvitaan taitavia insinöörejä kehittämään tekoälyjä, jotka ratkaisevat teknisiä ongelmia, mutta tarvitaan myös vähintään yhtä taitavia poliittisia päättäjiä, jotta koneoppimisen kehittymisen tuomat haasteetkin saadaan ratkaistua. Yhteiskunnat ovat selvinneet aikaisemmin suuristakin yhteiskunnallisista muutoksista, vaikeivat usein ilman muutokseen liittyviä kasvukipuja. Toivottavasti ihmiskunta osaa oppia historiasta, jotta tekoälyn ja koneoppimisen tuoma elintason parannus olisi kaikille saavutettavissa ja pahimmat skenaariot vältettäisiin.

## 5 YHTEENVETO

Tekoäly on koneoppimista käyttävä tietokonesysteemi, jolla pyritään erilaisten ongelmien ratkaisuun. Koneoppiminen voi hyödyntää erilaisia tekniikoita datan käsittelyyn ja lähestymistapoja mallin opettamiseen. Tässä työssä on käsitelty keinotekoisia neuroverkkoja mallina ja siihen liittyviä keskeisimpiä lähestymistapoja: ohjattua oppimista, ohjaamatonta oppimista ja vahvistusoppimista. Lisäksi on avattu ohjattuun oppimiseen liittyvää backpropagation-algoritmiksi kutsuttua oppimisalgoritmia.

Keinotekoiset neuroverkot ovat aivojen toiminnasta inspiraatioita ottava koneoppimisen yksi haara. Keinotekoiset neuroverkot ovat viime vuosikymmeninä kehittyneet huomattavasti ja useita kerroksia sisältävillä keinotekoisilla neuroverkoilla suoritettua koneoppimista sanotaan syväoppimiseksi. Keinotekoiset neuroverkot perustuvat toisiinsa yhteyksissä oleviin laskentapisteesiin, joita sanotaan neuroneiksi ja verkon laskenta ja oppiminen perustuvat neuroneiden ja niiden yhteyksien vapaiden parametrien muokkautumiseen paremmaksi verkon ollessa vuorovaikutuksessa sen oppimisympäristöön.

Syväoppimisen etu perinteisempään koneoppimiseen verrattuna on keinotekoisien neuroverkon kyky oppia haluttu malli ympäristöstä miltei suoraan sille syötettävästä datasta, jolloin suunnittelijan ei tarvitse tietää yhtä paljon ympäristöstä valmiiksi, kuten perinteisessä koneoppimisessa. Keinotekoinen neuroverkko toimiikin black box-mallina. Neuroverkko tosin tarvitsee huomattavasti enemmän dataa oppiakseen, mikä ei ole enää valtava haaste, kuten ennemmin. Nykyään dataa kerätään paljon enemmän miltei kaikesta, ja sitä on helpommin saatavilla tietotekniikan ja internetin kehityksen ansiosta.

Keinotekoisiiin neuroverkkoihin perustuvalla syväoppimisella on saatu ratkaistua ennen tekoälylle erittäin vaikeina pidettyjä ongelmia monella eri alalla. Tulevaisuudessa syväoppimisen ja koneoppimisen merkitys tulee näillä näkymin kasvamaan ja uusia sovelluksia kehitetään jatkuvasti.



## 6 LÄHDELUETTELO

Brownlee J., 2019, What is Deep Learning?, Machine Learning Mastery, [verkkodokumentti]  
 Saatavissa: <https://machinelearningmastery.com/what-is-deep-learning/>  
 [viitattu 28.12.2019]

Brynjolfsson E. & McAfee A., 2014, The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies, Norton & Company, Inc., 281 s. ISBN 978-0-393-24125-9.

Saatavissa:

[https://edisciplinas.usp.br/pluginfile.php/4312922/mod\\_resource/content/2/Erik%20-%20The%20Second%20Machine%20Age.pdf](https://edisciplinas.usp.br/pluginfile.php/4312922/mod_resource/content/2/Erik%20-%20The%20Second%20Machine%20Age.pdf) [viitattu 7.1.2020]

Hassabis D., Hubert T., Schrittwieser J., Silver D., 2018, AlphaZero: Shedding new light on chess, shogi, and Go, Deep Mind, [verkkodokumentti]

Saatavissa: <https://deepmind.com/blog/article/alphazero-shedding-new-light-grand-games-chess-shogi-and-go> [viitattu 28.12.2019]

Haykin S., 2009, Neural Networks and Learning Machines, 3. painos, New Jersey: Pearson Education Inc., 934 s. ISBN: 978-0-13-129376-2

Mueller J. P. & Massaron L., 2017, Machine Learning for dummies, John Wiley & Sons, Inc, 420 s. ISBN: 978-1-119-24551-3

Saatavissa: [http://lira.epac.to/DOCS-](http://lira.epac.to/DOCS-TECH/Algoritmi/Machine%20Learning%20For%20Dummies%20(John%20Paul%20Mueller%20-%20Luca%20Massaron).pdf)

[TECH/Algoritmi/Machine%20Learning%20For%20Dummies%20\(John%20Paul%20Mueller%20-%20Luca%20Massaron\).pdf](http://lira.epac.to/DOCS-TECH/Algoritmi/Machine%20Learning%20For%20Dummies%20(John%20Paul%20Mueller%20-%20Luca%20Massaron).pdf) [viitattu 9.1.2020]

Mäkelä I., 2019, Älykkäiden laskennallisten menetelmien soveltaminen teräksenvalmistusprosesseissa, Oulun yliopisto, 25 s.

Nielsen M., 2015, Neural Networks and Deep Learning, Determination Press

Saatavissa: <http://neuralnetworksanddeeplearning.com/index.html> [viitattu 4.1.2020]

Rosin P. L. & Fierens F., 1995, Improving Neural Network Generalisation, Institute for Remote Sensing Applications, Joint Research Centre, 3s.

Saatavissa:

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.48.3136&rep=rep1&type=pdf> [viitattu 14.1.2020]

Sanderson G., 2017, ”What is backpropagation really doing? | Deep learning, chapter 3”, 3Blue1Brown-YouTube kanava, [YouTube-video]

Saatavissa: <https://www.youtube.com/watch?v=Ilg3gGewQ5U> [viitattu 22.1.2020]

Saxén H. & Pettersson F., 2006, Method for the selection of inputs and structure of feedforward neural networks, Computers and chemical engineering 30, sivut 1038 - 1045.

Saxén H. & Pettersson F., 2007, Nonlinear prediction of the hot metal silicon content in the blast furnace, ISIJ International, sivut 1732-1737.

Silver D., 2016, Deep Reinforcement Learning, Deep Mind, [verkkodokumentti]

Saatavissa: <https://deepmind.com/blog/article/deep-reinforcement-learning> [viitattu 28.12.2019]

Theobald O., 2017, Machine Learning for Absolute Beginners, 52 s.

Saatavissa:

<http://120.107.155.180/download/MachineLearning/Machine%20Learning%20for%20Absolute%20Beginners.pdf> [viitattu 9.1.2020]