

VAASAN YLIOPISTO
TEKNILLINEN TIEDEKUNTA
TIETOTEKNIikka

Alexi Salo

NEUROVERKON KUVAAMINEN
VIEREKKYYSRELAATIOJÄRJESTELMÄLLÄ

Tietotekniikan
pro gradu -tutkielma

Multimediajärjestelmien ja teknisen viestinnän koulutusohjelma

VAASA 2016

SISÄLLYSLUETTELO**sivu**

JOHDANTO.....	6
1 KUVAAMISEN MENETELMIÄ.....	11
<u>1.1</u> Joukko.....	11
<u>1.2</u> Verkko.....	11
<u>1.3</u> Tietotyyppi.....	12
<u>1.4</u> 4-kerroksinen metamallihierarkia.....	12
2 VIEREKKYYSRELAATIOJÄRJESTELMÄT.....	14
<u>2.1</u> Käsitteet.....	14
<u>2.2</u> Yhdistäminen.....	17
<u>2.3</u> Vierekkyyssrelaatiojärjestelmä tietotyyppinä.....	18
3 NEUROVERKOT.....	19
<u>3.1</u> Neuroverkko.....	20
<u>3.1.1</u> Etenevä neuroverkko.....	20
<u>3.1.2</u> Takautuva neuroverkko.....	21
<u>3.2</u> Oppiminen.....	24
<u>3.3</u> Neuroverkko tietotyyppinä.....	24
4 NEUROVERKON KUVAAMINEN VIEREKKYYSRELAATIOJÄRJESTELMÄLLÄ.....	26
<u>4.1</u> Esimerkki: MLP.....	27
<u>4.2</u> Tietotyyppi.....	31
<u>4.3</u> Neuroverkon muodostaminen kuvauksesta.....	31
5 YHTEENVETO.....	33
6 JOHTOPÄÄTÖKSET.....	37

Symbolit ja lyhenteet

ARS	Adjacency Relation System
ARST	Adjacency Relation System associated with type defining sets
CWM	Common Warehouse Model
RBM	Restricted Boltzmann machine
SOM	Self-organizing map
UML	Unified Modeling Language
x_i	Elementti
A_i	Elementtijoukko
\mathfrak{A}	Elementtijoukkojen joukko
$(x_i, \{ y_i \})$	Relaatio y:stä x:ään; x:n viereisten elementtien joukko
R_{ij}	Relaatioiden joukko joukon A_i elementeistä joukon A_j elementteihin
\mathfrak{R}	Relaatiojoukkojen joukko
T_i	Tyyppi
\tilde{T}_{ij}	Tyyppijoukko; joukkojen A_i ja A_j vierekkyuden määrittelevä joukko
τ	Tyyppijoukkojen joukko
$\{ A_1 \dots A_n \} \setminus \{ A_j \}$	Joukkojen $A_1 \dots A_n$ joukko pois lukien joukko A_j
\in	Sisältyminen joukkoon
\neq	Eriarvoisuus
\emptyset	Tyhjä joukko

Kuviot

Kuvio 1. Perceptron. Sivu 21.

Kuvio 2. Hopfield-verkko. Sivu 22.

Kuvio 3. Rajoitettu Boltzmann-kone. Sivu 23.

Kuvio 4. Itseorganisoituva kartta. Sivu 24.

Kuvio 5. Multilayer Perceptron, MLP. Sivu 28.

Kuvio 6. Vierekkyyssrelaatiojärjestelmällä kuvattu neuroverkko. Sivu 29.

VAASAN YLIOPISTO**Teknillinen tiedekunta****Tekijä:****Aleksi Salo****Tutkielman nimi:****Neuroverkon kuvaaminen
vierekkyyssrelaatiojärjestelmällä****Ohjaajan nimi:****Merja Wanne****Tutkinto:****Kauppätieteiden maisteri****Ohjelma:****Multimediajärjestelmien ja teknisen viestinnän
koulutusohjelma****Pääaine:****Tietotekniikka****Opintojen aloitusvuosi:****2015****Tutkielman valmistumisvuosi:****2016****Sivumäärä: 41**

TIIVISTELMÄ

Tutkimusongelmana oli selvittää onko vierekkyyssrelaatiojärjestelmällä mahdollista kuvata neuroverkko. Hypoteesina on, että vierekkyyssrelaatiojärjestelmällä pystytään kuvaamaan tarkastikin erilaisia rakenteita sekä myös rakenteiden ominaisuuksia – myös neuroverkko. Vierekkyyssrelaatiojärjestelmällä kuvataan rakenteita formaalisti joukkojen avulla. Joukkomäärittelyt määrittävät myös relaatioita joukkojen alkioiden eli elementtien väleille, jotka voidaan esittää myös visuaalisesti verkonkaltaisena kuviona.

Neuroverkko on keinotekoinen hermoverkko, jonka malli on otettu aivojen tiedonkäsittelystä konnektionistisen näkemyksen mukaisesti. Neuroverkko koostuu neuroneista, jotka ovat yksinkertaisia prosessoreita, mutta joiden avulla voidaan käsitellä tehokkaasti moniulotteistakin tietoa. Neuronit jaetaan kerroksiin siten, että kussakin kerroksessa on useita neuroneja, jotka suorittavat prosessointinsa samanaikaisesti. Kerroksittain samanaikainen prosessointi tekee neuroverkosta hyvin rinnakkaissuoritteisen ja neuroverkon prosessointiin käyttämä aika voidaankin laskea kerrosten mukaan neuroverkon neuronien summan sijasta.

Vierekkyyssrelaatiojärjestelmä voidaan esittää, ja on yleensä esitetty joukkojen joukkona tai verkkomaisena kuviona jonka solmut on ryhmitelty. Neuroverkko esitetään usein verkon neuronin matemaattisena mallina tai verkkona, jonka neuronit on ryhmitelty kerroksiin. Vierekkyyssrelaatiojärjestelmällä kuvattu neuroverkko eroaa huomattavasti alkuperäisestä muodostaan, mikä tässä tutkielmassa käy ilmi. Vierekkyyssrelaatiojärjestelmällä kuvatussa neuroverkossa on kahden mallinnustason mallien piirteitä. Kuvaukseen sisältyy esimerkiksi relaatioiden joukot, jotka eivät muutu vaikka kuvattava neuroverkko vaihtuisi tai muuten muuttuisi. Toisaalta taas kuvaukseen sisältyy mm. kuvattavan neuroverkon neuronit, kaaret ja kerrokset siten että niiden lukumäärä käy ilmi, mikä on riippuvaista kuvattavasta neuroverkosta.

AVAINSANAT: vierekkyyssrelaatiojärjestelmä, neuroverkko, kuvaaminen

UNIVERSITY OF VAASA**Faculty of technology****Author:****Aleksi Salo****Topic of the Master's Thesis:****Defining neural network in
an adjacency relation system****Instructor:****Merja Wanne****Degree:****Master of Science in Economics and
Business Administration****Major:****Computer Science****Degree programme:****Degree Programme in Multimedia Sys-
tems and Technical Communication****Year of Entering the University:****2015****Year of Completing the Master's Thesis:** **2016****Pages: 41**

ABSTRACT

Aim of this thesis is to find out if and how neural networks can be defined in an adjacency relation system. Hypothesis is that this is possible since components of Adjacency Relation Systems are general in a way that allows definition of relations between practically arbitrary objects and types. Sets are used to define types in Adjacency Relation Systems. Sets are also used to define the relations between the objects. The types and the relations can then be presented in a graph-like figure.

Artificial neural network is an information processing system that mimics brain activity based on connectionism theory. Neurons of artificial neural network are simple processors which can process efficiently multidimensional data. The neurons are grouped in layers and the neurons in each layer process data simultaneously in a neural network. These parallel and concurrent processors make neural networks relatively efficient compared to serially processing systems.

An adjacency relation system is usually defined as set of sets and presented in a graph with groups of nodes. Neural network is often defined as mathematical model of neurons' function or represented as graph with neurons grouped in layers. When neural network is defined as an adjacency relation system, the network may be stored as an adjacency relation system and then restored to its original form. Defined adjacency relation system models instances but also classes or types as can be seen easily from the graph-like figure.

KEYWORDS: adjacency relation systems, neural network, modelling

JOHDANTO

Neuroverkot keksittiin ja niiden kehitys alkoi neurotieteen kehittyessä pisteeseen, jossa aivojen tietojenkäsittely pystyttiin selittämään hermosolujen yhteistoiminnan avulla. Neuroverkkojen tietojenkäsittelyn mallit perustuvat siten konnektionismiin. Vuonna 1943 Warren McCulloch ja Walter Pitts esittelivät hermosolun mallin – neuronin. McCullochin ja Pittsin mallin avulla voidaan ”laskea” yksittäisen neuronin toiminta (Bose & Liang 1996: 25).

Neuroverkkojen malli on saatu aivoista ja kuten ihminen oppii, myös neuroverkko voidaan opettaa. Donald Hebb esitti vuonna 1949, että oppiminen tapahtuu hermosolujen välisten yhteyksien vahvistuessa tai heikentyessä. Ensimmäisiä McCullochin ja Pittsin neuronin ja niin kutsutun Hebbin säännön yhdistäviä opetettavia neuroverkkoja oli perceptron, jonka esitteli Frank Rosenblatt 1950-luvun lopulla (Bose & Liang 1996: 120). Vaikka perceptron koostui jo useasta McCullochin ja Pittsin neuronista ja kahdesta kerroksesta, sen opetettavuus osoittautui rajalliseksi ja sai innostuksen jälkeen osakseen kovasti kritiikkiä 1960-luvulla. Perceptronin rajallisuuden ratkaisuksi tarjottiin neuroverkkoa useita perceptroneita rinnakkain ja useammassa peräkkäisessä kerroksessa. Tällaiselle neuroverkolle ei kuitenkaan ollut olemassa oppimisalgoritmia ja epäiltiin sellaisen olemassa olon mahdollisuutta. Useissa lähteissä (mm. Hertz et al. 1993: 6, Bose & Liang: 1996: 120) mainitaan tämän kritiikin aiheuttamaa kiinnostuksen suurta hiipumista neuroverkkojen tutkimusta kohtaan 1980-luvulle saakka. Hiipumisesta huolimatta tutkiminen jatkui 1970-luvulla keskittyen samankaltaisten syötteiden yhteyksien tutkimiseen (Hertz et al. 1993: 7).

1980-luvulle tultaessa ja kiinnostuksen taas kasvaessa keksittiin niin sanottuja rekurrentteja neuroverkkoja, joiden sisäinen toiminta saattoi olla kaoottisempaa tai vaikeasti hallittavampaa kuin perceptronin, joka oli ollut niin sanotusti yksisuuntainen neuroverkko. Rekurrentteja neuroverkkoja ovat esim. vuonna 1982 John Hopfieldin esittelemä ja hänen mukaansa nimetty Hopfield-verkko, vuonna 1989 Teuvo Kohosen esittelemä itseorganisoituva kartta (Zou et al. 2008) ja vuonna 1984 David Ackleyn, Geoffrey Hintonin ja Terrence Sejnowskin esittelemä stokastinen Boltzmann-kone ja myöhemmin rajoitettu stokastinen Boltzmann-kone. Vaikka neuroverkkojen rakenne ja toiminta tunnetaan hyvin, voi neuroverkkoja vielä nykyään pitää mustina laatikoina, koska niiden sisäisen tilan tulkinta on vaikeaa.

Tiedolla voidaan ajatella olevan rakenne, kun tiedolla tarkoitetaan jotakin laajempaa asiaa kuin yksittäistä, atomista tietoa. Yksittäinen, atominen tieto voi olla esimerkiksi nimi. Laajempi asia voisi olla vaikka ihminen, jonka yksi osa on hänen nimensä. Tällöin ihminen voitaisiin kuvata häneen liittyvien tietojen mukaisesti. Muita tietoja voisivat olla esimerkiksi vasemman silmän väri, kallon paksuus tai paksusuolen pinta-ala riippuen siitä mitä tai miten halutaan kuvata. Tiedolle muodostuu rakenne sitä mukaa, kun tietoja yhdistetään toisiinsa.

Rakenne muodostuu osistaan ja osiensa välisistä yhteyksistä - ilman yhteyksiä ei ole rakennetta ja ilman osia ei ole mitä yhdistää ja mitään mistä ylipäänsä muodostaa rakenne. Yhteyksien perusteella voidaan määritellä suhteet osien väleille eli miten osat liittyvät toisiinsa. Esimerkiksi kontekstista riippuen kirjalla ja koiralla ei ole välttämättä mitään yhteyttä toisiinsa, kun taas kirjalla ja kirjan sivulla olisi. Mutta jos kirja kertoo koirista, kirjan ja koiran välillä on olemassa yhteys kirjan aiheen kautta. Kahden asian yhteys voi olla mahdollista esittää useilla eri tavoilla tai eri asioiden kautta.

Tässä tutkielmassa kuvaamisella tarkoitetaan kohteen tarkkaa määrittelyä. Esittämisellä tarkoitetaan kuvauksen esittämistä esimerkiksi formaalisti matemaattisesti, visuaalisesti verkkomaisena kuviona, jonka solmut on ryhmitelty tai ohjelmointikielellä määriteltynä tietotyypinä. Mallintamisella tarkoitetaan kohteen osittaista kuvausta tai kuvauksen osittaista esittämistä – yksinkertaistettua mallia. Mallinnuksessa voidaan jättää huomiotta tarkoitukselle epäolennaisia tietoja. Esimerkiksi, neuroverkko on malli hermostosta ja neuroverkon neuronin malli hermoston hermosolusta – molemmat mallit ovat yksinkertaistettuja, tärkeiltä osiltaan kohdettaavia kuvauksia. Tässä tutkielmassa tavoitteena on kuvata neuroverkon rakenne ja rakenteen osien ominaisuudet. Neuroverkko on järjestelmä, joka saa tietoa syötteenä ja antaa tietoa tulosteena. Syötteenä annettu tuloste muodostaa takaisinkytkennän, jolloin järjestelmä käsittelee uudelleen jo käsittelemänsä tulosteen. Vierekkyyssrelaatiojärjestelmän ajatellaan tässä tutkielmassa tässä mielessä olevan enemmän kuvaus kuin järjestelmä.

Vierekkyyssrelaatiojärjestelmä on rakenne ja järjestelmä, jonka avulla voidaan kuvata tietojen kokonaisuus. Kokonaisuuksia ovat esimerkiksi jo mainitut ihminen ja koirakirja liittyvine osineen kuten kallon paksuus tai aihe. Vierekkyyssrelaatiojärjestelmässä kokonaisuus esitetään kahtena joukkona: erityyppisten tietojen eli elementtien joukkoina ja näiden tietojen välisten yhteyksien eli relaatioiden joukkoina. Vierekkyyssrelaatiojärjestelmä voidaan havainnollistaa ja esittää verkkomaisena, solmuistaan ryhmiteltynä ku-

viona (Wanne 1998, Töyli 2002, Töyli 2006, Nyrhilä & Mäenpää 2013b). Tässä työssä hahmotellaan myös vierekkyysrelaatiojärjestelmän esittämistä tietotyypinä.

Vierekkyysrelaatiojärjestelmän monimutkaisuus riippuu ensisijaisesti siihen sisältyvien tyyppien määrästä. Samasta kohteesta saatavien tyyppien määrä voi vaihdella riippuen mistä vinkkelistä kuvaus halutaan toteuttaa. Esimerkiksi lause voitaisiin kuvata sen sisältämien merkkien mukaan joko muodostamalla tyypit kirjaimista, numeroista ja erikoismerkeistä tai sitten lauseen jokaisesta merkistä, niin että lauseen kaikki samantyyppiset merkit muodostavat kukin yhden tyyppin. Suuri määrä tyyppijä ja erityisesti elementtejä tekee vierekkyysrelaatiojärjestelmällä muodostetun mallin selkeästä esittämisestä kuitenkin käytännössä haastavaa.

Neuroverkko on neuronien ja neuronien välisten yhteyksien eli kaarien suunnattu verkko. Sen tärkein ominaisuus on kyky oppia sekä myös ”muistaminen” ilman suurta erillistä tietovarastoa. Neuroverkon jokainen neuroni on yksinkertainen prosessori, joka syötteenä perusteella joko aktivoituu tai ei aktivoidu, eli lähettää tai ei lähetä impulssia lähteviä kaariaan pitkin seuraaville neuroneille. Jokaiselle kaarelle on määritelty painoarvo, jolla kuormitetaan seuraavaa neuronua. Neuroni aktivoituu ja lähettää impulssin, kun sille määritelty kynnyсарvo ylittyy kuormituksen seurauksena. Neuroverkkoja käsitellään luvussa 4. (Kokkarinen & Ala-Mutka 2002: 321–322.)

Neuroverkon malli on saatu biologiasta ja neurotieteestä – hermostosta. Hermosto välittää signaaleja eri kehonosista usein aivojen kautta muihin kehonosiin. Aivot ovat eräänlainen hermoston keskus, jossa hermoradat yhdistyvät toisiinsa. Aivojen tiedonkäsittelyn ominaisuuksia ovat muun muassa suuri rinnakkaisen laskennan ja prosessorien välisten yhteyksien määrä, binääritilat ja jatkuva-arvoiset muuttujat sekä oppiminen. Neuroverkon rinnakkaisessa tietojenkäsittelyssä sama tieto ikään kuin prosessoidaan useasta eri näkökulmasta siinä määrin kun kerroksessa on neuroneja. Silloin kun kaaria on kerroksissa jokaisesta neuronista jokaiseen neuroniin, yhdessä neuronissa tapahtuvassa prosessoinnissa otetaan huomioon myös muiden kerroksen neuronien syöte, eli neuronilla tavallaan on konteksti, jossa se suorittaa prosessointinsa. Neuronien tilat ovat aina binäärisiä siten, että ne joko lähettävät impulssin tai eivät. Aktivaatioarvot ja painot ovat jatkuva-arvoisia, jolloin ne voivat periaatteessa asettua äärettömän tarkasti. Oppiminen on neuronien ja neuronien välisten kaarien painojen muutosta. Toisin sanoen tiettyä tehtävää varten tietyt neuronien väliset yhteydet vahvistuvat tai heikentyvät. Mahdollis-

ta on myös että neuronin lähettää impulssin herkemmin tai vaihtoehtoisesti vaatii suurempaa kuormitusta. (Bose & Liang 1996: 5–8.)

Neuroverkon toiminta jaetaan yleensä kahteen vaiheeseen – opettamiseen ja itse käyttöön. Opettaminen on käytännössä neuronien välisten kaarien painojen ja mahdollisesti neuronien aktivaatioarvojen muuttamista. Kaaren painon muuttuminen voidaan ajatella tarkoittavan neuronien välisen yhteyden vahvistumista tai heikentymistä. Oppimisvaiheen jälkeen neuroverkko on valmis käytettäväksi, eli sille voidaan antaa syöte, jonka neuroverkko käsittelee oppimansa mukaisesti. Biologisen hermoverkon voidaan ajatella olevan jatkuvasti käytössä ja samalla olevan koko ajan oppiva, minkä vuoksi esimerkiksi ihminen voi oppia jatkuvasti omista tekemisistään syy-seuraus -suhteiden ymmärtämisen kautta.

Opettaminen voidaan toteuttaa ohjatusti tai ohjaamattomasti. Opetettaessa neuroverkolle annetaan syötteitä. Ohjatusti opetettaessa neuroverkon kaarien painoja ja neuronien aktivaatioarvoja muutetaan siten että neuroverkon tuloste muodostuu halutunlaiseksi. Ohjaamattomasti opetettaessa neuroverkko asettuu johonkin tilaan saamiensa syötteiden perusteella. Erästä ohjatun ja ohjaamattoman oppimisen välimuotoa kutsutaan vahvistusoppimiseksi, jolloin neuroverkolle kerrotaan että tulos oli oikea. Vääriä tulosteita ei kuitenkaan korjata ohjatun oppimisen tavoin (Bose & Liang 1996: 193).

Neuroverkon monimutkaisuus riippuu siihen kuuluvien neuronien ja neuroneja yhdistävien kaarien määrästä ja suunnasta. Tietyyntyyppisissä neuroverkoissa on silmukoita, jotka vaikuttavat esimerkiksi neuroverkon niin kutsuttuun asettumiseen opetettaessa. Toisaalta neuronien toiminta on aina melko yksinkertaista. Yksi neuronin laskee yksinkertaisen totuusarvon syötteidensä perusteella niin kutsulla aktivaatiofunktiolla. Totuusarvon perusteella neuronin joko kuormittaa tai ei kuormita etenevien kaariensa päissä olevia neuroneja binäärisesti tai dynaamisemmin suhteessa omaan kuormittumiseensa esimerkiksi jonkin logistisen käyrän mukaisesti. Esimerkiksi, kaksikerroksisen neuroverkon, jonka ensimmäisellä kerroksella on kaksi neuronin ja toisella yksi neuronin, toisen kerroksen neuronin aktivoituu, kun kuormitus ylittää sen aktivaatioarvon. Jos toisen kerroksen neuronin aktivaatioarvo on 1 ja molempien siihen tulevien kaarien paino 0,5 ja molemmat ensimmäisen kerroksen neuronit aktivoituvat, kuormittuu toisen kerroksen neuronin yhteensä molempien kaarien painolla, jolloin se aktivoituu, koska sen aktivaatioarvo saavutetaan. Dynaamisemmassa tapauksessa, jos aktivaatioarvo olisi 0, kaarista toisen paino 0,4, aktivaatiofunktio on logistinen käyrä ja molemmat ensimmäisen ker-

roksen neuronit aktivoituisivat, voisi toisen kerroksen neuronin kuormittaa seuraavia neuroneja 0,9-kertaisesti lähtevien kaariensa painolla, jos sillä niitä olisi.

Tiettyjen toimintojen suoritus on aivoissa keskittynyt tietylle alueelle. Tällaista aivojen ja hermoston osaa voidaan pitää erikoistuneena johonkin tehtävään. Monien toimintojen suorittamiseen tarvitaan kuitenkin useita alueita aivoissa, jotka kaikki voivat olla erikoistuneista jonkin toiminnon suorittamiseen. Jonkin erikoistuneen hermoston osan vioittuessa ja siinä siinä syntyneen toiminnan lakatessa, toiminta voi palautua tai siirtyä muihin toimintaan alun perin liittyneisiin aivojen osiin (Bose & Liang 1996: 5–8).

Tutkimuskohteena ovat vierekkyysrelaatiojärjestelmät. Tarkoituksena on kartoittaa vierekkyysrelaatiojärjestelmän sovellusmahdollisuuksia testaamalla vierekkyysrelaatiojärjestelmää neuroverkolla. Tutkimuskysymyksenä on voidaanko vierekkyysrelaatiojärjestelmällä kuvata neuroverkko siten kuin neuroverkko tässä tutkielmassa on määritelty eli viiden vierekkyysrelaatiojärjestelmän tyyppin avulla. Neuroverkon toimintaan liittyvät olennaisesti neuronien ja kaarien ominaisuudet. Näiden ominaisuuksien kuvaaminen on myös tutkimuksen kohteena. Lisäksi johtopäätöksissä kiinnitetään huomiota vierekkyysrelaatiojärjestelmän ajallisen kuvaamisen rajoitteisiin.

Luvussa 3 esitellään vierekkyysrelaatiojärjestelmien teoriaa ja luvussa 4 neuroverkkojen yleistä rakennetta sekä lopuksi luvussa 6 kuvataan MLP-neuroverkko vierekkyysrelaatiojärjestelmällä. Ensin kuitenkin luvussa 2 käydään lyhyesti läpi johdonmukaisuuden kannalta tarpeellisia käsitteitä kuten joukko ja verkko matemaattisina konsepteina. Joukkoja käytetään vierekkyysrelaatiojärjestelmien teoriassa vierekkyysrelaatiojärjestelmän määrittelyyn. Tässä tutkielmassa neuroverkko määritellään verkkona, toisin kuin yleensä yhden neuronin matemaattisena mallina: aktivaatiofunktiona. Joukon ja verkon lisäksi esitellään tietojenkäsittelyssä käytettävä muuttujan määrittelyssä käytettävä tietotyyppi vierekkyysrelaatiojärjestelmälle ja neuroverkolle. Esiteltävä tietotyyppi on niin sanottu monimutkainen (engl. complex) tietotyyppi eli se voi sisältää useita eri tietotyyppisiä ja arvoja. Tietotyypeille ei tässä tutkielmassa määritellä käsittelyä. Lisäksi luvussa 2 esitellään UML-mallinnusjärjestelmä, jota verrataan lyhyesti vierekkyysrelaatiojärjestelmällä muodostettuun neuroverkon kuvaukseen.

1 KUVAAMISEN MENETELMIÄ

Yleisesti, kuvaamistapoja on useita, esimerkiksi symbolit objekteille tai subjekteille, operaattorit toiminnalle tai kuvio mallille. Vierekkyyssrelaatiojärjestelmä kuvataan joukkojen avulla, joista voi myös muodostaa elementtien verkon. Tässä työssä vierekkyyssrelaatiojärjestelmän rakennetta havainnollistetaan esittämällä vierekkyyssrelaatiojärjestelmän kuvaus kuviona, jossa yhdistyvät joukot ja elementtien verkko.

1.1 Joukko

Joukko sisältää alkioita tai se voi olla tyhjä. Alkiot ovat objekteja, joilla on yleensä jokin yhteinen ominaisuus. Toisin sanoen, samankaltaiset objektit kuuluvat samaan joukkoon. Joukkoon kuuluva objekti voi olla esimerkiksi vierekkyyssrelaatiojärjestelmän elementti tai toinen joukko. Formaalisti joukot määritellään joukko-opin aksioomilla. Joukkojen avulla voidaan määritellä muun muassa matematiikan konsepteja ja matemaattisia käsitteitä. (Hrbacek & Jech 1999.)

Joukkoa voidaan merkitä matematiikassa formaalisti isolla kirjaimella A ja objektia eli alkioita tai elementtiä pienellä kirjaimella x . Kun x kuuluu joukkoon A , merkitään $x \in A$. Joukon A sisältö eli joukkoon kuuluvat objektit voidaan kuvata ja esittää formaalisti merkitsemällä $A = \{ x \}$. Jos elementti x ei kuulu joukkoon A , merkitään $x \notin A$. Elementtien x ja y järjestettyä paria merkitään (x, y) ja järjestämätöntä joukkoa $\{ x, y \}$ tai $\{ y, x \}$.

1.2 Verkko

Verkko G muodostuu joukoista solmuja, joita merkitään symbolilla V ja kaaria, joita merkitään symbolilla E , eli $G = (V, E)$. Verkkoteoriassa yksinkertaisen ja yleisen määritelmän mukaan verkko on joukot solmuja ja solmuja yhdistäviä kaaria (Voloshin 2009: 2). Kaari voi olla suunnattu tai suuntauton. Suunnattu kaari on alkupäässä olevan solmun kannalta lähtevä kaari ja loppupäässä olevan solmun kannalta tuleva kaari. Solmuilla on lähtöaste, joka kertoo kuinka monta kaarta solmusta yhteensä lähtee sekä tuloaste, joka kertoo kuinka monta kaarta solmuun tulee. Suuntaamattoman kaaren ajatellaan sekä tulevan että lähtevän solmusta.

Verkkoteoriassa kahden solmun sanotaan olevan vierekkäisiä, kun niitä yhdistää kaari ja kahden kaaren sanotaan olevan vierekkäisiä, jos liittyvät samaan solmuun (Voloshin 2009: 3). Vierekkyyssrelaatiojärjestelmissä vierekkyydellä on eri merkitys. Verkkoteoriassa voidaan vierekkäistä solmuja joskus kutsua myös naapuriksi ja kaikkien vierekkäisten solmujen joukkoa naapurustoksi (Voloshin 2009: 3). Vierekkyyssrelaatiojärjestelmässä vierekkyydellä tarkoitetaan ennemminkin solmun ja kaaren yhteyttä. Kaaren päissä olevien solmujen sanotaan olevan vierekkäisiä kaaren kautta.

1.3 Tietotyyppi

Tietojenkäsittelyssä tietojen käsittely toteutetaan muuttujien avulla. Muuttuja on yleensä symboli-arvo –pari, jossa symboli on muuttujan tunniste ja arvo on muuttujan arvo. Muuttujalle on yleensä määritelty myös tyyppi, jonka mukaan muuttujan arvo tulkitaan. Esimerkiksi tietokoneen muistissa muuttujan arvo on binäärinen numeroarvo, mutta jos arvo on kirjaintyyppinen, voidaan arvoa vastaavan numeron sijasta esittää arvoa vastaava kirjain. Monimutkainen (engl. complex) tietotyyppi voi sisältää useita muuttujia. Tällaista tietotyyppiä kutsutaan yleensä structiksi, esimerkiksi C-ohjelmointikielessä.

Tietorakenne on useita muuttujia hallitusti käsittelevä järjestelmä. Tietorakenteisiin liittyy yleensä myös rakenteen lisäksi rakenteen käsittelyt kuten lisäämisen, poistamisen tai etsimisen. Tietorakenteita ovat muun muassa puu, keko, kasa ja verkko. Tässä tutkielmassa esitetään luvuissa 3.3 ARS-tietotyyppi ja 4.3 Neuroverkko-tietotyyppi C-ohjelmointikielen struct-tietotyypin avulla.

1.4 4-kerroksinen metamallihierarkia

Malli on mallintamalla muodostettu yleistys mallinnuksen kohteesta. Malli voidaan muodostaa esimerkiksi rakenteesta tai toiminnasta. Kohde voidaan mallintaa eri tasoilla – usein yleisesti, mutta riittävän tarkasti. Mallin mallia kutsutaan metamalliksi (Fowler 2005: 9).

Järjestelmä voidaan kuvata eri tasoilla muodostamalla mallista yleisemmän tai tarkemman mallin. 4-kerroksinen metamallihierarkia muodostuu neljästä tasosta: M0, M1, M2 ja M3. M0-taso kuvaa järjestelmää tarkimmin ja M3-taso yleisimmin. M0-tasolla esittää esimerkiksi jokainen suoritukseen aikainen instanssi (engl. snapshot), kun taas M1-tasolla yleisesti, yhtenä ainoana instanssina – instanssien mallina.

Yleensä järjestelmän malli muodostetaan M1-tasolla. Järjestelmää, ja mallia, yleisemmin kuvaavia M2- ja M3-tason malleja kutsutaan metamalliksi ja metametamalliksi. Meta- ja metametamallit mallintavat järjestelmän sijasta enemmänkin järjestelmästä jo muodostettua mallia. M3-tason metametamalleja voidaan muodostaa MOFilla (Meta Object Facility), M2-tason metamalleja UML:llä ja CWM:llä (Common Warehouse Metamodel). (UML: 17.)

2 VIEREKKYYSRELAATIOJÄRJESTELMÄT

Vierekkyysrelaatiojärjestelmä (Adjacency Relation System, ARS) on joukkojen joukkojen pari $(\mathfrak{A}, \mathfrak{R})$, eli joukkoihin \mathfrak{A} ja \mathfrak{R} sisältyy joukkoja. Vierekkyysrelaatiojärjestelmän osia ovat elementit, relaatiot ja joukot. Elementit kuuluvat joukkoihin ja elementti yhdistetään saman tai toisen joukon elementtiin relaatiolla. Joukko \mathfrak{A} on elementtien joukko ja \mathfrak{R} on relaatioiden joukko. Tässä tutkielmassa, toivottavasti selkeyttäen, joukkoa joka sisältää elementtejä kutsutaan elementtijoukoksi ja joukkoa joka sisältää tyyppisiä tyyppijoukoksi. ARSia johon on liitetty vierekkyuden määrittelevät joukot eli tyyppijoukot kutsutaan ARSTiksi (Adjacency Relation System associated with adjacency defining sets) ja jota merkitään $(\mathfrak{A}, \mathfrak{R}, \tau)$. Joukko τ sisältää tyyppijoukot. (Wanne 1998.)

Vierekkyysrelaatiojärjestelmiä on sovellettu puolirakenteisen (semistructured) tiedon käsittelyssä. Töyli (2002) on muodostanut vierekkyysmallin (Adjacency Model, AM), joka perustuu vierekkyysrelaatiojärjestelmien teoriaan sekä algoritmin puolirakenteisen datan jäsentämiseen (2006) vierekkyysmallin avulla. Mäenpää ja Nyrhilä ovat muuntaaneet semanttisten linkkien verkon (semantic link network, SLN) ARSiksi (2013a) ja kehittäneet menetelmän, jolla voidaan kuvata semanttista dataa vierekkyysrelaatiojärjestelmällä (2013b). Mäenpää ja Wanne (2015) ovat tutkineet viereisyysmallin ja relaatiomallin (Relational Model) yhteneväisyyksiä. Mäenpää (2015) on tutkinut vierekkyysmallin hyödyntämistä graafianalyysissä.

ARSin osia ovat elementit, elementtien väliset kaaret ja joukot, joihin elementit sisältyvät. Kukin joukko määrittelee tyyppin joukkoon sisältyville elementeille.

2.1 Käsitteet

Vierekkyysrelaatiojärjestelmien teorian peruskäsitteitä ovat vierekkyys, ARS, ARST, yksiselitteinen (engl. unique) ARST sekä pätevä (engl. valid) relaatiokombinaatio. ARS muodostuu parista $(\mathfrak{A}, \mathfrak{R})$. Kun ARS-rakenteeseen lisätään tyyppijoukot τ , kutsutaan muodostunutta rakennetta ARSTiksi. Yhden joukon elementit ovat yhden tyyppisiä elementtejä. Pätevä relaatiokombinaatio voidaan määritellä yksiselitteiselle ARSTille. Relaatiokombinaatio on jokin valittu joukko yksiselitteisen vierekkyysrelaatiojärjestelmän relaatioita – relaatioiden kombinaatio.

Vierekkyys

Relaatio määrittelee vierekkyuden kahden elementin välille. Elementin x viereisiä elementtejä merkitään joukolla $Adj(x)$. Elementin $y \in A_j$ kuulumista joukkoon $Adj(x)$ merkitään $Adj(x) = \{ y \}$ tai $y \in Adj(x)$. Elementin ei sanota olevan vierekkäinen itselleen; jos $x \in A_i$ niin $x \notin Adj_i(x)$. Elementtien sanotaan olevan toistensa naapureita, jos ne ovat samasta joukosta ja lisäksi niiden välillä on relaatio tai ne ovat vierekkäisiä tyyppijoukkonsa kautta.

Vierekkyysrelaatiojärjestelmä on symmetrinen, kun kaikkien elementtien, joiden välillä on relaatio, on kaksi relaatiota vastakkaisiin suuntiin. Toisin sanoen, kaksi elementtiä on aina vierekkäisiä toisilleen. Epäsymmetrisessä vierekkyysrelaatiojärjestelmässä on mahdollista määrittellä kahdesta elementistä ainoastaan toinen vierekkäiseksi toiselle.

Tässä tutkielmassa sanotaan että kaikista tyyppijoukoista joukkojen A_i ja A_j elementtien tyyppijoukkoon kuuluvat ne tyypit, jotka määrittelevät vierekkyuden joukkojen A_i ja A_j elementeille. Joukkojen A_i ja A_j elementtien vierekkyuden määrittelevää joukkoa merkitään \tilde{T}_{ij} .

ARS

ARS koostuu joukkojen joukosta \mathfrak{A} ja relaatioiden joukosta \mathfrak{R} . \mathfrak{A} sisältää järjestelmän erilliset ja epätyhjät joukot $\{ A_1, A_2, \dots, A_n \}$. \mathfrak{R} sisältää vierekkyysrelaatiojärjestelmän relaatiot R_{ij} , kun $i, j \in \{ 1, 2, \dots, n \}$. R_{ij} suunnattujen relaatioiden joukko joukosta A_j joukkoon A_i . (Wanne 1998: 9–10.)

ARST

ARSin joukkojen $A_1 \dots A_n$ elementit edustavat tyyppiä $T_1 \dots T_n$. Joukon A_i elementtien tyyppiä merkitään T_i . ARST koostuu ARS:n joukkojen \mathfrak{A} ja \mathfrak{R} lisäksi vierekkyuden määrittelevien tyyppien joukosta τ . Joukkojen A_i ja A_j vierekkyys määritellään kyseisten joukkojen yhteisten vierekkäisten joukkojen elementtien kautta. Vierekkyuden määrittelevä joukko on joukon $\{ A_1 \dots A_n \} \setminus \{ A_i, A_j \}$ jokin osajoukko. Joukkoon τ ei siis voi kuulua joukot A_i ja A_j . Joukkojen A_i ja A_j vierekkyuden määrittelevää tyyppijouk-

koa merkitään \tilde{T}_{ij} , joka on joukkojen A_i ja A_j yhteiset vierekkäiset joukot. \tilde{T}_{ij} voi olla tyhjä ja voi olla että $i = j$. (Wanne 1998: 11–12.)

Yksiselitteinen ARST

Vierekkyyden määrittelevät joukot τ valitaan ARSissa määriteltyjen relaatioiden perusteella. ARST on yksiselitteinen (engl. unique), jos vierekkyuden määrittelevät joukot määrittelevät vierekkyuden kaikille kahden tai saman joukon elementeille. Toisin sanoen kaikki elementit $x \in A_i$ ja $y \in A_j$ ovat vierekkäisiä ainoastaan \tilde{T}_{ij} kautta. (Wanne 1998: 13.)

Pätevä relaatiokombinaatio

Relaatiokombinaatio on relaatiotyyppien joukko. Relaatioiden R_{ij} tyyppiä – yhtä relaatiotyyppiä – merkitään $T_i \rightarrow T_j$. Relaatiokombinaatio S on pätevä (engl. valid) jos siihen liittyvä ARST on ainut muodostuvissa oleva yksiselitteinen ARST. Jos yksiselitteinen ARST $(\mathfrak{A}, \mathfrak{R}, \tau)$ on yksiselitteinen myös ARSTina $(\mathfrak{A}, \mathfrak{R}', \tau)$, ei relaatiokombinaatio ole pätevä. (Wanne 1998: 13.)

Epäsyyntisyys

Epäsyyntiset eli suunnatut relaatiot voivat muodostaa elementtien väleille monimutkaisia liitoksia, koska niitä voi olla paitsi eri joukkojen myös saman joukon elementtien kesken. Epäsyyntisyys voi viitata esimerkiksi käsitteellisesti hierarkkiseen yhteyteen vaikkapa ihmisen ja lapsen välillä – suunnattuun relaatioon lapsesta ihmiseen: lapsi on ihminen, mutta ihminen ei välttämättä lapsi. Suunnatun relaation alkupäässä on lapsi, jonka määrittänyt relaation loppupäässä, ihmisellä, mutta ei toisinpäin.

Yhdistetty ARS

Yhden elementtijoukon ARS, $\mathfrak{A} = \{ A \}$, sanotaan olevan yhdistetty, jos mitkä tahansa kaksi joukon elementtiä kuuluu samaan sarjaan vierekkäisiä elementtejä. Toisin sanoen mistä tahansa elementistä on mahdollista päästä mihin tahansa elementtiin siirtymällä relaatiota pitkin viereiseksi määriteltyyn elementtiin.

2.2 Yhdistäminen

Vierekkyyssrelaatiojärjestelmien teoriassa yhdistämisellä tarkoitetaan vierekkyyssrelaatiojärjestelmän eräänlaista yksinkertaistamista sulauttamalla elementtejä yhteen. Yhdistämistä varten elementit ovat ns. väritettyjä – elementit ovat ryhmitelty värinsä mukaan, mutta niitä ei ole varsinaisesti jaettu uusiin joukkoihin. Elementtijoukon kahdesta eri elementistä toinen on yhdistettävissä toiseen, kun elementit ovat samanväriset ja kuuluvat samaan vierekkäisten elementtien jaksoon eli sarjaan tai ketjuun. Yhdistäminen suoritetaan saman joukon elementeille eli samantyyppisille elementeille. (Wanne 1998: 69–73.)

Wanne (1998) määrittelee ja analysoi kolme erilaista käsittelyä elementtien yhdistämiseksi. Käsittelyt muodostuvat kahdesta, järjestyksessä peräkkäisestä, vaiheesta: etsintä- ja yhdistämisvaiheesta. Etsintävaiheessa etsitään yhdistettävissä olevia elementtejä ja yhdistämisvaiheessa löydetty yhdistettävissä olevat elementit yhdistetään.

Jokainen käsittely alkaa ja jatkuu jos käsittelemättömiä elementtejä löytyy sekä päättyy, kun kaikki elementit on käsitelty. Käsittelyt eroavat siten, että etsintävaiheessa ne etsivät joko yhden yhdistettävissä olevan elementin, kaikki vierekkäiset yhdistettävissä olevat elementit tai kaikki yhdistettävissä olevat elementit. Yhdistämisvaiheessa löydetty elementit yhdistetään. Yhdistämisvaiheen jälkeen jatketaan etsintävaiheeseen jos käsittelemättömiä elementtejä on jäljellä, muuten suoritus lopetetaan. (Wanne 1998: 74–75.)

Em. käsittelyjen tärkein ero toisiinsa nähden on etsintä- ja yhdistämisvaiheiden suorituskertojen määrässä. Ts. yhdellä etsintävaiheen suorituksella voidaan etsiä joko yksi tai useampi elementti kerrallaan. Etsittäessä useampia elementtejä kerralla vaiheen suorituskertoja on vähemmän, koska löydetty elementit eivät enää ole aloitus- ja jatkamisedon mukaisesti käsittelemättömiä. Vaiheiden suorituskerroista ei käy ilmi algoritmien tehokkuutta. Käsittelyjen analysointia varten Wanne (1998: 75) määrittelee tarkemmat käsittelyt joista etsintä- ja yhdistämisvaiheet koostuvat, ja joiden avulla analysointi on mahdollista.

2.3 Vierekkyyssrelaatiojärjestelmä tietotyypinä

ARS määritellään joukkojen parina. ARS voidaan esittää C-ohjelmointi kielen tietotyypinä, joka muodostuu tietotyypeistä jotka kuvaavat elementtien ja relaatioiden joukkoja:

```
struct Elementti {
    void* elementti;
}

struct Relaatio {
    struct Elementti elementti;
    struct Elementti *viereinen;
}

struct Joukko {
    unsigned int index;
    struct Elementti *elementti;
}

struct ARS {
    struct Joukko *joukot;
    struct Relaatio *relaatiot;
}
```

ARS-tietotyyppi sisältää taulukot (array) joukoille ja relaatioille. Joukko-tietotyyppi sisältää tiedot järjestysnumerostaan sekä sisältämistään elementeistä, jotka voidaan määritellä taulukkoon. Elementti-tietotyyppi sisältää yleisen viittauksen muistipaikkaan. Relaatio-tietotyyppi sisältää ARS:n määritelmien mukaisesti elementin sekä sen viereiset elementit.

3 NEUROVERKOT

Neuroverkko eli keinotekoinen hermoverkko on tiedonkäsittelyjärjestelmä, jonka toimintamalli on saatu aivojen toiminnasta konnektionistisen, eli rinnakkaisen ja hajautetun tiedonkäsittelyn teorian mukaisesti (Zou et al. 2008, Bishop & Mitchell 1991). Neuroverkko koostuu yksiköistä eli prosessoreista – keinotekoisista neuroneista – jotka on järjestetty kerroksiin. Jokainen keinotekoinen neuroni on yksinkertainen prosessori, josta on yhteyksiä muihin neuroverkon neuroneihin (Bishop & Mitchell 1991). Neuroniprosessori joko aktivoituu, toisin sanoen lähettää impulssin saamansa syötteen perusteella, tai vaihtoehtoisesti ei aktivoidu (McCulloch & Pitts 1943). Aktivoituessaan neuroni siten ikään kuin jatkaa neuroverkon läpi kulkevaa impulssia tai sitten pysäyttää sen. Biologisessa hermoverkossa neuroni voi lähettää myös impulssin, joka vaimentaa tai kokonaan estää seuraavia neuroneja aktivoitumasta. Aivoissa on noin 100 miljardia neuronit ja jokaisesta neuronista on yhteyksiä 1000–10000 neuroniin (Uhrig 1995). Sen perusteella voidaan ajatella, että aivojen hermosto olisi neuroverkkona enemmän pitkä kuin leveä, kun jokaisesta neuronista on seuraavan kerroksen neuroneihin 1000–1000 yhteyttä, yksi kuhunkin. Yhtä pitkä kuin yhtä leveä neuroverkko tarkoittaisi korkeintaan 10000^2 eli 100 miljoonaa neuronit, sillä kerroksessa voisi olla korkeintaan 10000 neuronit. 100 miljardin neuronin neuroverkkoon tarvitaan siten lisää kerroksia. Pällekkäiset tai rinnakkaiset kerrokset ovat hyödyttömiä, ellei niihin liity erillisiä syöte- tai tulostekerroksia.

Neuroverkko koostuu neuroneista ja neuronien välisistä kaarista. Neuronit on järjestetty peräkkäisiin kerroksiin. Jokaisessa kerroksessa on vähintään yksi neuroni ja jokainen neuroni on täsmälleen yhdessä kerroksessa. Neuroverkon ensimmäinen kerros on syötekerros ja viimeisen tulostekerros. Laskettaessa neuroverkon kerroksia syötekerrosta ei yleensä lasketa mukaan, sillä se ei prosessoi tietoa vaan ainoastaan lukee sisään (Hertz et al. 1993:90). Tässä tutkielmassa syötekerros lasketaan neuroverkon kerrokseksi kuvauksen yhdenmukaistamiseksi. Toisaalta syötekerroksen yksiköiden voidaan ajatella olevan myös opetettavissa (engl. fixed), kun taas muiden kerrosten yksiköt ovat opetettavissa (Bose & Liang 1996: 120). Syötekerroksesta lähteviä kaaria voidaan opettaa. Syöte- ja tulostekerroksen neuronit eroavat muista neuroneista siten, että niihin ei sekä tule ja lähde kaaria. Syötekerroksen neuroneista on yhteys ainoastaan eteneviä kaaria, koska syötekerrosta edeltäviä kerroksia ei ole. Tulostekerroksen neuroneihin ainoastaan tulee kaaria, koska tulostekerroksen jälkeen ei ole kerroksia. (Zou et al. 2008.)

Tässä tutkielmassa läpikäytävät neuroverkot ovat siinä mielessä aina suunnattuja, että neuroverkoilla on syötekerros ja tulostekerros. Syötekerroksen kautta luetaan neuroverkkoon tietoa eli syötettä. Tulostekerroksen kautta neuroverkko antaa tietoa eli tulosteen. Kuvattavat neuroverkot jaetaan eteneviin (engl. feedforward) ja takautuviin (engl. feedback (Zou et al.), recurrent (Bose & Liang)) neuroverkkoihin. Etenevän neuroverkon kaikki kaaret ovat kerroksittain eteneviä. Takautuvan neuroverkon kaaret voivat olla lisäksi kerroksittain takautuvia tai kerroksessa pysyviä eli sivuttaisia.

Eteneviä neuroverkkoja ovat esim. perceptron (kuvio 1) ja MLP (Multi-Layer Perceptron). Itseorganisoidut kartat (self-organizing maps, Kohonen maps) (kuvio 2) luokitellaan kirjallisuudessa yleensä omaksi tyypikseen tai takautuvaksi neuroverkoksi. Itseorganisoidun kartan piilokerroksen neuronit on järjestetty kaksiulotteisesti tasolle. Muita takautuvia neuroverkkoja ovat esim. Hopfield-verkko (kuvio 3), Boltzmann-kone (Boltzmann machine, BM) ja rajoitettu Boltzmann-kone (Restricted Boltzmann machine, RBM) (kuvio 4). Esim. perceptron, Hopfield-verkko ja Boltzmann-koneet ovat Bishopin ja Mitchellin mukaan (1991) eräitä neuroverkkojen päätyyppejä. Zou et al. (2008) luokittelevat neuroverkot yleisemmin eteneviin ja takautuviin neuroverkkoihin.

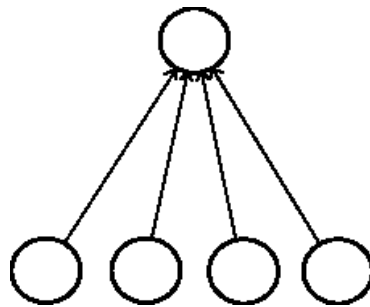
3.1 Neuroverkko

Yleisesti, neuroverkko G koostuu joukkojen joukosta $\{ K, V, E \}$, missä K on kerrosten joukko, V on neuronien joukko ja E on kaarien joukko. Kerrokset $K_{0..n}$ sisältävät neuronit $v_i \in K_k$, missä kerroksen järjestysluku $k = 0..n$ ja neuronin järjestysluku $i = 0..n$. Kaari e_{ij} on neuronien i ja j välinen kaari, missä i on tuloneuroni ja j on lähtöneuroni, ts. kaari e_{ij} tulee neuronin i . Lisäksi jokaiselle neuronille on määritelty kynnyisarvo $\theta \in \mathbb{R}$ ja kaarelle painoarvo $\omega \in \mathbb{R}$.

3.1.1. Etenevä neuroverkko

Etenevässä (feedforward) tai eteenpäinsyöttävässä (Kokkarinen & Ala-Mutka 2002: 321) neuroverkossa on ainoastaan kerroksittain eteneviä kaaria. Neuronit ovat järjestettyinä neuroverkon kerrokseen siten, että jokainen neuroni on täsmälleen yhdessä kerroksessa ja jokaisen kaaren suunta on kohti tulostekerrosta. (Hertz et al. 1993: 90.)

Etenevässä neuroverkossa, jokaisen kaaren e_{ij} neuronit v_i ja v_j ovat peräkkäisissä kerroksissa siten, että $v_i \in K_k$ ja $v_j \in K_{k-1}$, kun $k \neq 0$. Kerroksen K_{k-1} neuronista v_j on kaari e_{ij} kaikkiin neuroneihin v , kun $v \in K_k$.



Kuvio 1. Perceptron.

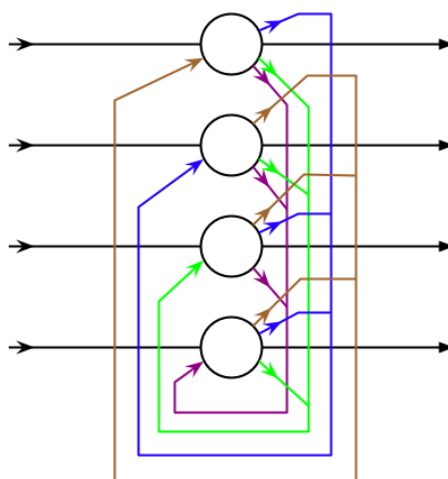
Perceptron on rakenteeltaan yksinkertainen, kahdesta prosessointikerroksesta ja lisäksi syötekerroksesta muodostuva neuroverkko (Bose & Liang 1996: 120—121). Sen esitteli ensimmäisenä Frank Rosenblatt 1950-luvun lopulla. Monikerroksinen perceptron (Multi-Layer Perceptron, MLP) koostuu ikään kuin rinnakkaisista ja peräkkäisistä perceptroneista kuitenkin siten, että syötekerroksia on vain yksi – ensimmäinen kerros. MLP:ssä rinnakkaiset perceptronit jakavat viereisten kerrosten neuronit.

3.1.2. Takautuva neuroverkko

Takautuvassa (feedback, recurrent) neuroverkossa voi olla etenevien kaarien lisäksi takautuvia tai sivuttaisia kaaria, jolloin neuroverkkoon syntyy silmukoita (Kokkarinen & Ala-Mutka 2002: 321). Takautuvan kaaren tuloneuronin on lähtöneuronin kerrosta aiemmassa kerroksessa. Sivuttaisen kaaren lähtö- ja tuloneuronit ovat samassa kerroksessa. Syötekerroksesta lähtee ainoastaan eteneviä kaaria. Tulostekerroksesta ei lähde kaaria.

Takautuvassa neuroverkossa tuloste voi muodostua jollakin aikavälillä – ”takautuvan neuroverkon käyttäytyminen on ajallista” (Bose & Liang 1996: 283) - toisin kuin tässä tutkielmassa määritellyissä etenevissä neuroverkoissa, joissa yhdestä syötteestä muodostuu yksi tuloste aina yhdellä ajanhetkellä. Takautuvassa neuroverkossa olevien silmukoiden seura

uksena silmukan kautta kulkeva informaatio jää ikään kuin jälkeen vain etenevästä informaatiosta. Toisaalta taas eteenpäin kerroksia ohittavat yhteydet nopeuttavat prosessointia ja voivat synnyttää eräänlaista etukäteistulostetta. Silmukoiden ja ohitusten kautta syötteestä saatu informaatio voi vaikuttaa sitä seuraavista tai sitä edeltävistä syötteistä saatuun informaatioon, jos syötteiden sisältämä informaatio on neuroverkossa samanaikaisesti. Takautuvassa neuroverkossa sama syöte voi synnyttää erilaisia tulosteita.



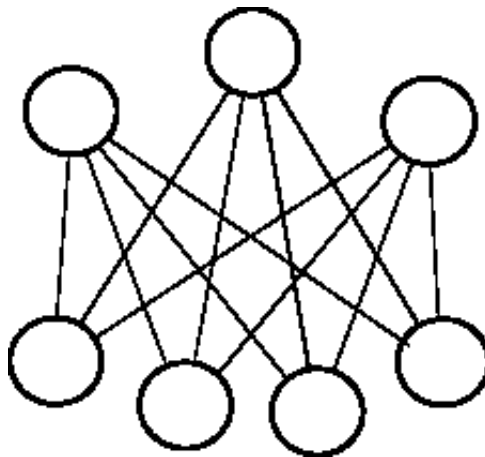
Kuvio 2. Hopfield-verkko. (CC.)

Hopfield-verkossa kerroksen neuronien väleillä on symmetriset sivuttaiset kaaret muihin saman kerroksen neuroneihin (Bose & Liang 1996: 289), ts. jokaisesta neuronista on suunnattu kaari jokaiseen toiseen saman kerroksen neuroniin. Hopfield-verkko on kehitetty vuonna 1982 ja nimetty kehittäjänsä John Hopfieldin mukaan. Symmetriset kaaret voidaan korvata kahdella vastakkaisuuntaisella kaarella, joilla on aina täsmälleen samat painot. Hopfield-verkko on nk. assosiatiivinen neuroverkko. Termi assosiatiivinen viittaa assosiatiiviseen muistiin. Assosiatiivinen muisti yhdistää – assosioi – informaation johonkin aikaisemmin oppimaansa informaatioon. Hopfield-verkon tapauksessa verkko asettuu (engl converge) sen aikaa kun informaatiota prosessoidaan kerroksessa.

Boltzmann-kone (Boltzmann machine, BM) koostuu kahden tyyppisistä neuroneista: näkyvistä (visible) ja piilotetuista (hidden). Jokaisesta neuronista voi olla kaari mihintahansa muuhun neuroniin. Boltzmann-koneessa saman kerroksen näkyvät neuronit toimivat sekä syöte- että tulosteneuroneina. Boltzmann-kone Hopfield-verkon laajennoksena koostuu näkyvistä Hopfield-verkon neuroneista, joista on kaaret kerroksen piilotettuihin neuroneihin. Boltzmann-koneen piiloneuroneihin voi tallentua syötteen sisä-

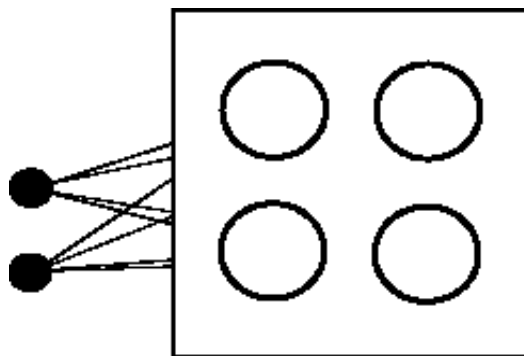
nen kuvaus, ts. syötteestä saadun informaation tila kyseisessä kerroksessa. (Bose & Liang 1996:324.)

Rajoitettu Boltzmann-kone (Restricted Boltzmann machine, RBM) (kuvio 3) on kaksijakoinen Boltzmann-kone ja koostuu myös näkyvistä ja piilotetuista neuroneista. Kaksijakoisuus rajoittaa kaaria siten, että jokaisesta neuronista voi olla kaari ainoastaan erityyppiseen neuroniin, eli näkyvästä piilotettuun ja päinvastoin.



Kuvio 3. Rajoitettu Boltzmann-kone (RBM).

Itseorganisoituva kartta eräs ohjaamattomasti oppiva neuroverkko. Itseorganisoituvan kartan on kehittänyt Teuvo Kohonen vuonna 1982. Sen neuronit voidaan järjestää janelle tai tasoon (Hertz et al. 1993: 236). Itseorganisoituvan kartan sen neuronin, joka eniten muistuttaa syötettä, kaaren painoa muutetaan paremmin syötettä vastaavaksi. Lisäksi sen viereisten neuronien kaarien painoja muutetaan vain hieman paremmin syötettä vastaavaksi. (Bose & Liang 1996: 361.)



Kuvio 4. Itseorganisoituva kartta

3.2 Oppiminen

Neuroverkon oppimisalgoritmit voidaan jakaa kahteen pääluokkaan algoritmin toiminnan perusteella (Zou et al. 2008: 19). Oppimisalgoritmit luokitellaan ohjatun oppimisen algoritmeihin ja ohjaamaton oppimisen algoritmeihin. Ohjatusti opetettuna neuroverkolle kerrotaan oikea tulos eli mihin verkon tulokseen sen on päästävä. Neuroverkko ohjataan oikeaan tulokseen syötteen ja tulosteen perusteella. Ohjaamattomasti opetettuna neuroverkon tulee itse löytää oikea tulos syötteiden perusteella. Ts. neuroverkko asettuu johonkin tilaan oppimansa perusteella, ja tilan oletetaan olevan oikea. Käytännössä neuroverkon oppiminen on kaarien painojen ja mahdollisesti neuronien aktivaatioarvojen muuttumista. Painojen tai aktivaatioarvojen muuttuessa tietyt neuronien väliset yhteydet vahvistuvat tai heikentyvät suhteellisesti toisiinsa nähden.

Neuroverkko opetetaan syötteiden joukolla, jota kutsutaan opetusjoukoksi (training set). Neuroverkon oppimisella tarkoitetaan esim. sitä, että neuroverkko pystyy johtamaan opetusmateriaalinsa perusteella oikeita tuloksia myös syötteistä, joilla sitä ei ole opetettu. Etenevä neuroverkko voidaan opettaa nk. backpropagation-algoritmillä. Algoritmi vertaa syntynyttä syötettä haluttuun syötteeseen, jonka perusteella kaarien painoja ja mahdollisesti neuronien aktivaatioarvoja muutetaan lasketun virheen verran halutun oppimisnopeuden eli virheenkorjauskertoimen mukaisesti.

3.3 Neuroverkko tietotyypinä

Neuroverkko voidaan kuvata tietotyypinä seuraavasti:

```
struct Neuroni {
    double aktivaatioarvo;
    struct Kaari* etenevä;
    struct Kaari* takautuva;
}

struct Kaari {
    double paino;
    struct Neuroni* aiempi;
    struct Neuroni* jaljempi;
}
```

```
struct Kerros {  
    struct Neuronit* neuronit;  
}  
  
struct Neuroverkko {  
    struct Kerros* kerrokset;  
}
```

Neuroverkko-struct määrittelee neuroverkon kerrokset Kerros-structilla. Kerros-struct puolestaan määrittelee kerroksen neuronit Neuronit-structin avulla. Neuronit – joka sijaitsee neuroverkon kerroksessa – määrittelee neuroniin tulevat ja siitä lähtevät kaaret Kaari-structin avulla sekä itselleen aktivaatioarvon. Kaikilla neuroneilla ei ole sekä eteneviä että takautuvia kaaria. Kaari-structilla määritellään kaikki neuroverkon kaaret. Jokaiselle kaarelle määritellään kaaren alkupään sekä loppupään neuronit Neuronit-structilla.

4 NEUROVERKON KUVAAMINEN VIEREKKYYSRELAATIOJÄRJESTELMÄLLÄ

Tämän työn tarkoituksena on selvittää onko vierekkyysrelaatiojärjestelmällä mahdollista kuvata neuroverkko. Aiemmissa luvuissa on käyty läpi vierekkyysrelaatiojärjestelmä ja neuroverkko sekä esitelty niiden ominaisuuksia. Lisäksi myös joukon ja verkon ominaisuuksia esiteltiin lyhyesti, sillä seuraavat kuvaukset ja määritelmät yritetään muodostaa niiden pohjalta vierekkyysrelaatiojärjestelmien teorian mukaisesti. Näiden lisäksi on esitelty myös tietotyyppi sekä mallintamiseen liittyvät mallinnustasot. Vierekkyysrelaatiojärjestelmiä ja neuroverkkoja käsittelevissä luvuissa on muodostettu tietotyypit, joiden avulla voidaan toteuttaa ARS ja neuroverkko ATK-ohjelmassa.

Kuvauksessa halutaan nähdä neuronit, kaaret, kerrokset, neuronien aktivaatioarvot ja kaarien painot, joten muodostetaan joukot:

$$\mathfrak{A} = \{ A_1, A_2, A_3, A_4, A_5 \}$$

Joukko A_1 sisältää neuroverkon neuronit: $A_1 = \{ v_1, v_2, \dots, v_n \}$.

Joukko A_2 sisältää neuroverkon kaaret: $A_2 = \{ e_1, e_2, \dots, e_m \}$.

Joukko A_3 sisältää kerrokset: $A_3 = \{ k_1, k_2, \dots, k_l \}$.

Joukko A_4 sisältää kaarien painot: $A_4 = \{ p_1, p_2, \dots, p_k \}$.

Joukko A_5 sisältää neuronien aktivaatioarvot: $A_5 = \{ t_1, t_2, \dots, t_j \}$.

Neuroverkosta tiedetään, että neuroni kuuluu kerrokseen, neuronit yhdistetään toisiinsa kaarilla, neuroneilla on aktivaatioarvot ja kaarilla painot, joten saadaan seuraavat relaatiot:

$$\mathfrak{R} = \{ R_{12}, R_{13}, R_{15}, R_{21}, R_{24}, R_{31}, R_{42}, R_{51} \}$$

Joukko \mathfrak{R} muodostuu neuronien ja kaarien välisistä yhteyksistä R_{12} ja R_{21} , neuronien ja joukkojen välisistä yhteyksistä R_{13} ja R_{31} , kaarien ja kaarien painojen välisistä yhteyksistä R_{24} ja R_{42} sekä neuronien ja neuronien aktivaatioarvojen välisistä yhteyksistä R_{15} ja R_{51} . Lisäksi muodostetaan relaatiokombinaatio, joka kuvaa neuroverkkoa: neuroni kuuluu kerrokseen (1, 3), neuroni ja kaari muodostavat verkon ja ovat yhtäläillä riippuvaisia toisistaan (1, 2) ja (2, 1), paino on kaaren ominaisuus (2, 4) ja aktivaatioarvo on neuronin ominaisuus (1, 5).

$$S = \{ (1, 3), (1, 2), (2, 1), (2, 4), (1, 5) \} = \{ (1, 2), (1, 3), (1, 5), (2, 1), (2, 4) \}$$

4.1 Esimerkki: MLP

Kuvataan MLP-neuroverkko (kuvio 5), jolla on kolme neuronia syötekerroksessa (neuronit v_1, v_2, v_3), kaksi neuronia ensimmäisessä piilokerroksessa (neuronit v_4, v_5), kolme neuronia toisessa piilokerroksessa (neuronit v_6, v_7, v_8) ja yksi tulostoneuroni (neuronit v_9).

$$\mathcal{A} = \{ A_1, A_2, A_3, A_4, A_5 \}$$

$$A_1 = \{ v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9 \}$$

$$A_2 = \{ e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15} \}$$

$$A_3 = \{ k_1, k_2, k_3, k_4 \}$$

$$A_4 = \{ p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15} \}$$

$$A_5 = \{ t_1, t_2, t_3, t_4, t_5, t_6 \}$$

Em. joukkojen välisten epätyhjien relaatioiden joukko $\mathcal{R} = \{ R_{12}, R_{13}, R_{15}, R_{21}, R_{24}, R_{31}, R_{42}, R_{51} \}$

$$R_{12} = \{ (v_1, \{e_1, e_2\}), (v_2, \{e_3, e_4\}), (v_3, \{e_5, e_6\}), (v_4, \{e_1, e_3, e_5, e_7, e_8, e_9\}), (v_5, \{e_2, e_4, e_6, e_{10}, e_{11}, e_{12}\}), (v_6, \{e_7, e_{10}, e_{13}\}), (v_7, \{e_8, e_{11}, e_{14}\}), (v_8, \{e_9, e_{12}, e_{15}\}), (v_9, \{e_7, e_{11}, e_{15}\}) \}$$

$$R_{21} = \{ (e_1, \{v_1, v_4\}), (e_2, \{v_1, v_5\}), (e_3, \{v_2, v_4\}), (e_4, \{v_2, v_5\}), (e_5, \{v_3, v_4\}), (e_6, \{v_3, v_5\}), (e_7, \{v_4, v_6\}), (e_8, \{v_4, v_7\}), (e_9, \{v_4, v_8\}), (e_{10}, \{v_5, v_6\}), (e_{11}, \{v_5, v_7\}), (e_{12}, \{v_5, v_8\}), (e_{13}, \{v_6, v_9\}), (e_{14}, \{v_7, v_9\}), (e_{15}, \{v_8, v_9\}) \}$$

$$R_{13} = \{ (v_1, \{k_1\}), (v_2, \{k_1\}), (v_3, \{k_1\}), (v_4, \{k_2\}), (v_5, \{k_2\}), (v_6, \{k_3\}), (v_7, \{k_3\}), (v_8, \{k_3\}), (v_9, \{k_4\}) \}$$

$$R_{31} = \{ (k_1, \{v_1, v_2, v_3\}), (k_2, \{v_4, v_5\}), (k_3, \{v_6, v_7, v_8\}), (k_4, \{v_9\}) \}$$

$$R_{24} = \{ (e_1, \{p_1\}), (e_2, \{p_2\}), (e_3, \{p_3\}), (e_4, \{p_4\}), (e_5, \{p_5\}), (e_6, \{p_6\}), (e_7, \{p_7\}), (e_8, \{p_8\}), (e_9, \{p_9\}), (e_{10}, \{p_{10}\}), (e_{11}, \{p_{11}\}), (e_{12}, \{p_{12}\}), (e_{13}, \{p_{13}\}), (e_{14}, \{p_{14}\}), (e_{15}, \{p_{15}\}) \}$$

$$R_{42} = \{ (p_1, \{ e_1 \}), (p_2, \{ e_2 \}), (p_3, \{ e_3 \}), (p_4, \{ e_4 \}), (p_5, \{ e_5 \}), (p_6, \{ e_6 \}), (p_7, \{ e_7 \}), (p_8, \{ e_8 \}), (p_9, \{ e_9 \}), (p_{10}, \{ e_{10} \}), (p_{11}, \{ e_{11} \}), (p_{12}, \{ e_{12} \}), (p_{13}, \{ e_{13} \}), (p_{14}, \{ e_{14} \}), (p_{15}, \{ e_{15} \}) \}$$

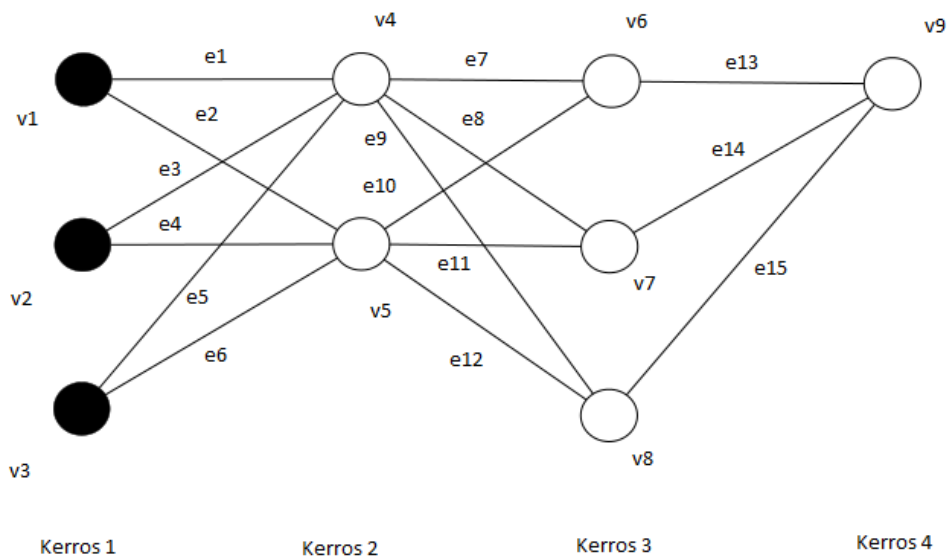
$$R_{15} = \{ (v_1, \emptyset), (v_2, \emptyset), (v_3, \emptyset), (v_4, \{ t_1 \}), (v_5, \{ t_2 \}), (v_6, \{ t_3 \}), (v_7, \{ t_4 \}), (v_8, \{ t_5 \}), (v_9, \{ t_6 \}) \}$$

$$R_{51} = \{ (t_1, \{ v_4 \}), (t_2, \{ v_5 \}), (t_3, \{ v_6 \}), (t_4, \{ v_7 \}), (t_5, \{ v_8 \}), (t_6, \{ v_9 \}) \}$$

$$R_{11} = R_{14} = R_{22} = R_{23} = R_{25} = R_{32} = R_{33} = R_{34} = R_{35} = \emptyset$$

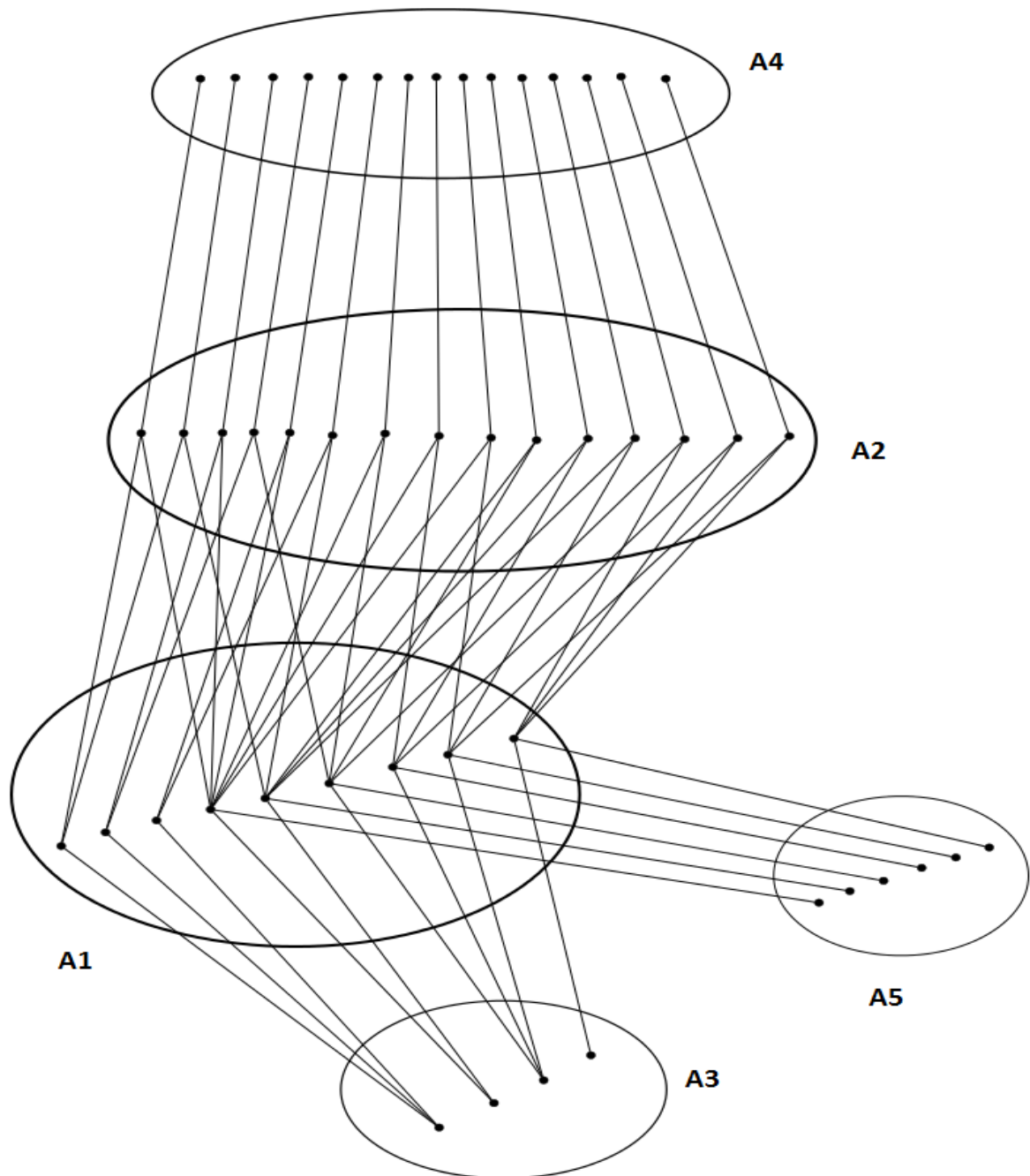
$$R_{41} = R_{43} = R_{44} = R_{45} = R_{52} = R_{53} = R_{54} = R_{55} = \emptyset$$

Jokaiset jonkin relaation päissä olevat kaksi elementtiä ovat toistensa viereisiä, joten ARS on symmetrinen. Esim. elementit e_1 ja e_2 ovat elementin v_1 viereisiä kuten on määritelty relaatioiden joukossa R_{12} ja elementti v_1 on viereinen elementeille e_1 ja e_2 kuten on määritelty relaatioiden joukossa R_{21} .



Kuvio 5. Multilayer Perceptron, MLP.

MLP-ARS on esitetty verkon ja joukkojen avulla kuviossa 6. Kuviosta käy selkeämmin ilmi elementtien väliset relaatiot. Elementit ovat kuviossa joukoissa kasvavassa järjestyksessä vasemmalta oikealle.



Kuvio 6. Vierekkyyssrelaatiojärjestelmällä kuvattu neuroverkko.

MLP-ARST on määritelmän mukaan yksiselitteinen (Wanne 1998: 12) seuraavilla tyyppijoukoilla eli vierekkyuden määrittelevillä joukoilla:

$$\tau = \{ \begin{aligned} &\tilde{T}_{11} = \{ T_2, T_3 \}, \\ &\tilde{T}_{14} = \tilde{T}_{41} = \{ T_2 \}, \\ &\tilde{T}_{22} = \tilde{T}_{23} = \tilde{T}_{32} = \tilde{T}_{25} = \tilde{T}_{52} = \{ T_1 \}, \\ &\tilde{T}_{12} = \tilde{T}_{13} = \tilde{T}_{15} = \tilde{T}_{21} = \tilde{T}_{24} = \tilde{T}_{33} = \tilde{T}_{34} = \tilde{T}_{35} = \emptyset, \\ &\tilde{T}_{42} = \tilde{T}_{43} = \tilde{T}_{44} = \tilde{T}_{45} = \tilde{T}_{51} = \tilde{T}_{53} = \tilde{T}_{54} = \tilde{T}_{55} = \emptyset \end{aligned} \}$$

Määritelmän mukaan yksiselitteisessä ARSTissa on mukana kaikki mahdolliset vierekkyuden määrittelevät joukot. Esimerkiksi kaikilla joukon A_1 elementeillä on yhteinen elementti joukoissa A_2 ja A_3 . Joukkojen A_2 ja A_5 elementeillä on yhteinen elementti joukossa A_1 , mutta joukkojen A_3 ja A_5 kaikilla elementeillä ei ole yhteistä elementtiä.

Joukkoa τ voidaan tulkita siten, että se muodostaa yhteyksiä toisilleen etäisiltäkin vaikuttavien tyyppien väleille. Epätyhjä joukko \tilde{T}_{ij} kertoo, että yhteys i - ja j -tyyppisten elementtien välillä on mahdollisesti olemassa – tyyppien elementit voivat olla vierekkäisiä. Yhteys on olemassa eli elementit ovat vierekkäisiä, jos elementeille on olemassa yhteinen vierekkyuden määrittelevän tyyppin elementti. Toisaalta, vierekkäisten elementtien avulla voidaan löytää tietty vierekkyuden määrittelevän tyyppin elementti.

Esimerkiksi, joukon \tilde{T}_{11} mukaan kahdella neuronilla voidaan määritellä kerros tai kaari. Esimerkin MLP:n kuvauksessa neuronien avulla määrittyvä kerros on kerros, johon molemmat eli vierekkäiset neuronit kuuluvat. Määrittyvä kaari taas on MLP:ssä kahden neuronin välinen kaari. Kahden neuronin suoran tai epäsuorankin yhteyden mahdollinen olemassaolo voi olla intuitiivisesti helppo hahmottaa, ehkä toisin kuin esimerkiksi neuronin ja kaaren painon välinen yhteys. Joukot \tilde{T}_{14} ja \tilde{T}_{41} määrittävät mahdollisen vierekkyuden neuronille ja kaaren painolle. Vierekkyys määritetään T_2 -elementin eli kaaren kautta. Neuronin ja kaaren painon avulla voidaan siten määrittää esimerkiksi mihin neuroniin kaari liittyy – mistä se lähtee tai tulee. Esimerkin MLP:ssä kaaren tulo- ja lähtöneuronin voidaan määrittää esimerkiksi neuronien indeksejä vertailemalla – kaaren suunta on etenevässä neuroverkossa pienemmästä indeksistä isompaan.

4.2 Tietotyyppi

Kappaleissa 3.3 ja 4.3 on kuvattu mahdolliset tavat määrittellä ARS ja neuroverkko tietotyyppinä C-ohjelmointikielen struct-tietotyypin avulla. Jotta neuroverkko-rakenne kuvaisi vierekkyyssrelaatiojärjestelmällä kuvattua neuroverkkoa, tulee paino ja aktivaatioarvo irroittaa esitellyistä Kaari- ja Neuron-structeista ja muodostaa niistä omat tietotyyppinsä, jolloin ne voidaan kuvata vierekkyyssrelaatiojärjestelmän elementteinä.

Metamalli kuvaa mallia, ja malli kuvaa suoritettavia instancesja. Tässä tapauksessa ARS-tietotyypin voidaan ajatella vastaavan metamallia, koska se kuvaa tiivistetysti neuroverkon tietotyypin. ARS-tietotyyppiin tulisi pystyä lukemaan tai lataamaan sisään neuroverkon tietotyypit, mahdollisesti yleisintä määriteltyä neuroverkon tietotyyppiä, eli Neuroverkko-tietotyyppiä, lukuun ottamatta.

4.3 Neuroverkon muodostaminen kuvauksesta

Neuroverkko tulisi pystyä koostamaan kuvauksensa elementeistä, sillä se vastaa kohdetaan yksi-yhteen. Kuvauksesta voidaan käyttää siihen sisältyvää verkkoa apuna elementtien algoritmisessa läpikäynnissä. Verkoille tiedetään olevan useita esimerkiksi syvyys- ja leveyssuuntaisia läpikäymisalgoritmeja.

Kuviota 6, jossa neuroverkko on kuvattu vierekkyyssrelaatiojärjestelmänä, voidaan käyttää apuna koostamisalgoritmillemme tarpeellisten sijoitusten ja tarkistusten läpi käynnin hahmottamisessa. Molemmissa esiteltävissä algoritmeissa käydään elementtejä läpi järjestyksessä ja toistuvasti siirtyen seuraavaan elementtiin, kunnes kaikkissa elementeissä on vierailtu:

1. Asetetaan neuronin aktivaatioarvo
2. Asetetaan neuronin kerros
3. Asetetaan neuronille kaari
4. Tarkistetaan onko kaarelle asetettu paino
 - a. Asetetaan kaarelle paino

Algoritmi voidaan myös toteuttaa käymällä edellisen algoritmin vaiheessa kolme lisäämässä kaari kaaren toisessa päässä olevalle neuroneille, asettamalla kaaren paino sekä asettamalla kerroksen kaikki neuronit:

1. Asetetaan neuronin aktivaatioarvo
2. Tarkastetaan vielä asettamaton kerros
 - a. Asetetaan neuronin kerros
 - b. Asetetaan kerroksen muut neuronit
3. Tarkastetaan vielä asettamaton kaari
 - a. Asetetaan neuronille kaari
 - b. Asetetaan kaarelle paino
 - c. Asetetaan ko. kaari kaaren toiselle neuronille

Tietotyyppien käsittelyssä voidaan hyödyntää jälkimmäistä käsittelyä. Toteutuksessa neuroverkko tallennettaisiin ARS-tietotyyppiin. Neuroverkko tulisi pystyä myös palautamaan ARS-tietotyyppistä alkuperäiseen muotoonsa.

5 YHTEENVETO

Tässä tutkielmassa käsiteltiin rakenteen kuvaamista vierekkyyssrelaatiojärjestelmällä. Kuvauksen kohteena oli neuroverkko. Vierekkyyssrelaatiojärjestelmällä pystytään kuvaamaan neuroverkon rakenne ja myös rakenteen osien ominaisuuksia. Neuroverkon toiminta ei käy ilmi tässä työssä muodostetusta vierekkyyssrelaatiojärjestelmästä, mutta toisaalta neuroverkon sisäinen tila tietyllä hetkellä voi olla mahdollista saada selville ominaisuuksien avulla. Neuroverkon rakenne pystyttiin kuvaamaan siten yleisesti, että erityyppisiä neuroverkkoja kuvattaessa vierekkyyssrelaatiojärjestelmään ei muodostu uusia tyyppisiä vaikkakin elementtien määrät ja relaatiot voivat vaihdella.

Vierekkyyssrelaatiojärjestelmä on elementtijoukkojen ja relaatioiden joukkojen avulla määritelty kuvaus järjestelmästä (tietyllä ajan hetkellä). Relaatiot yhdistävät elementit ja määrittävät vierekkyydet muodostaen myös toisaalta verkon. Jokainen elementtijoukko määrittää uuden tyyppin, jota joukkoon kuuluvat elementit edustavat. Neuroverkon kuvausta varten määriteltiin rakenteeksi kolme komponenttia: neuronit, kaaret ja kerrokset ja kaksi ominaisuutta: neuronin aktivaatioarvo ja kaaren painoarvo. Komponenteista ja ominaisuuksista saatiin vierekkyyssrelaatiojärjestelmään viisi tyyppiä.

Erityyppisten elementtien vierekkyyden määrittämiseen tyyppijoukon avulla. Yksiselitteisessä vierekkyyssrelaatiojärjestelmässä tyyppijoukkojen on sisällettävä kaikki vierekkyyttä määrittävät tyypit. Tyyppijoukoista ei siis voi puuttua niihin sopivia tyyppisiä, ilman että yksiselitteisyyden ehto rikkoutuisi. Yksiselitteiselle ARSTille voidaan valita relaatiokombinaatioita. Relaatiokombinaatio muodostetaan tyyppien välisistä relaatioista ja se on pätevä kun se perustuu relaatioille, joista on muodostettu ainoa mahdollinen yksiselitteinen ARST. Toisin sanoen, päteviä relaatiokombinaatioita voidaan muodostaa vain tietojen kokonaisuudelle, joka on määritelty yksiselitteisesti, ainutlaatuisesti ja täydellisesti.

Neuroverkot on luokiteltu kahteen luokkaan: eteneviin ja takautuviin neuroverkkoihin. Etenevä ja takautuva neuroverkko eroavat toisistaan siten, että etenevässä neuroverkossa neuronien väliset yhteydet ovat ainoastaan kerroksittain eteneviä kun taas takautuvassa neuroverkossa yhteydet voivat olla lisäksi takautuvia tai sivuttaisia eli kerroksessa pysyviä. Etenevä ja takautuva neuroverkko koostuvat samoista komponenteista ja ominaisuuksista. Yhdensuuntaisten yhteyksien neuroverkko on todennäköisesti helpommin hallittavissa kuin suunnaltaan kerroksittain vaihtelevien yhteyksien neuroverkko. Neu-

roverkon alku ja loppu ovat syöte- ja tulostekerrokset. Takautuvaan neuroverkkoon voi syntyä neuroverkon sisäisiä loputtomia silmukoita eli yksi annettu syöte voi aktivoida loputtomasti neuroneja. Oppiminen on neuroverkon tärkeimpiä ominaisuuksia ja se tapahtuu muuttamalla yhteyksien painoja ja neuronien aktivaatioarvoja. Painojen ja aktivaatioarvojen muutosten seurauksena neuronien aktivoituminen muuttuu. Oppiminen voi olla jatkuvaa ja loputonta ja se voi olla joko ohjattua tai ohjaamatonta.

ARSille ja neuroverkolle on esitetty tietotyypit. Tietotyypit voidaan toteuttaa ohjelmoimalla, mutta niistä puuttuvat yleensä tietorakenteisiin kuuluvat käsittelyt, joilla tietorakenteen sisältämää tietoa hallitaan. Tietotyyppejä voidaan käyttää myös tiedon tallentamiseen samoin kuin vierekkyyssrelaatiojärjestelmään määritettyjä tietoja voidaan säilyttää vierekkyyssrelaatiojärjestelmän rakenteessa. Vierekkyyssrelaatiojärjestelmään tai tietotyyppiin tallennettu tieto voidaan palauttaa tai ottaa käyttöön käymällä tallennetut tiedot algoritmisesti läpi. Toisin sanoen, tallennettu tieto on luettavissa algoritmisesti, toisaalta myös tallennettavissa. Toteutusten kannalta saattaa olla tärkeää tietää, että mistä aloittaa eli mikä tieto on ensimmäinen, jolloin elementtien järjestyksen tai vierekkyyden määrittäminen joukon sisällä voi olla tarpeellista.

Kävi ilmi, että vierekkyyssrelaatiojärjestelmäkuvauksessa on UML-mallinnusjärjestelmän kahden eri tason mallin piirteitä. Kuvauksesta hahmottuu M1-tason malli neuroverkosta, mutta samalla siitä käy ilmi instansseja eli M0-tason malli. Esimerkiksi neuroverkosta saadut viisi elementtien tyyppiä ovat samat kaikille neuroverkoille huolimatta siitä onko neuroverkko etenevä vai takautuva. Myös se että paljonko yhteyksiä tai elementtejä on, ei muuta vierekkyyssrelaatiojärjestelmän tyyppien lukumäärää. Erityistä kahden eri tason mallin esittämisessä samassa kuvauksessa on juuri siinä, että yleensä niitä ei pystytä esittämään tai ei esitetä samoissa kuvaajissa.

Vierekkyyssrelaatiojärjestelmän voi ajatella kuvaavan neuroverkkoa tietyllä ajan hetkellä, vaikka vierekkyyssrelaatiojärjestelmään ei liity ajallisuutta ilmaisevaa komponenttia. Vierekkyyssrelaatiojärjestelmä kuvaa neuroverkon rakennetta, mutta ei sen toimintaa – ajallisesti tai muutenkaan, mutta ominaisuuksien avulla voidaan kuvata neuroverkon tilaa tietyllä ajan hetkellä. Muodostamalla useita tilan kuvauksia olisi mahdollista seurata neuroverkon rakenteen muuttumista ja mahdollisesti ominaisuuksien kautta myös sisäisen tilan muuttumista. Tilan muuttumista kuvaavan aikakomponentin olisi oltava vähintään järjestystä ilmaiseva, jotta tilojen kuvausten järjestys voidaan tietää ja esimerkiksi muuttuminen esittää oikeassa järjestyksessä. Aikakomponentin lisääminen

vierekkyyssrelaatiojärjestelmään tulee vaikuttaa vierekkyyssrelaatiojärjestelmän kaikkiin elementteihin siinä tapauksessa, että vierekkyyssrelaatiojärjestelmän elementtien halutaan olevan kuvaushetkellä samassa ajassa. Diskreetin aikamuuttujan lisääminen olisi suhteellisen yksinkertaista, jolloin vierekkyyssrelaatiojärjestelmästä muodostettaisiin tilojen kuvauksia, useilla ajan hetkillä.

Jos vierekkyyssrelaatiojärjestelmään lisätään elementti tai vierekkyyssrelaatiojärjestelmään määritellään uusi tyyppi, ei vierekkyyssrelaatiojärjestelmä ehkä säily yksiselitteisenä ja valittu relaatiokombinaatio pätevänä. Uuden elementin lisäämisen jälkeen tulee varmistua ARSTin yksiselitteisyydestä ja mahdollisesti muodostaa uusi pätevä relaatiokombinaatio. Uuden elementin lisäämisen edellytyksenä on luonnollisesti tunnistaa uusi elementti, jotta sitä voidaan ylipäänsä käsitellä. Lisäksi on tunnistettava uuden elementin tyyppi, eli joukko johon elementti on lisättävä tai mahdollisesti tarve uudelle joukolle ja lisättävä elementti siihen. Elementti on myös yhdistettävä relaatioin muihin sille viereisiin elementteihin. Elementtien poistamisen yhteydessä tulee hävittää ylimääräiset relaatiot ja tyhjät joukot.

Edellä mainitun kaltainen vierekkyyssrelaatiojärjestelmää päivittävän tai korjaavan käsittelyn analysoimiseksi on määriteltävä alkeisoperaatiot, joiden avulla käsittely toteutetaan. Ensimmäisessä vaiheessa suoritettava uuden elementin tunnistus ja luokittelu on mahdollisesti olla monimutkainen tehtävä, johon esimerkiksi neuroverkko saattaa soveltua hyvin, sillä se rakenteensa vuoksi saattaa sietää suhteellisen hyvin ns. melua eli epäolennaista tietoa.

Sensorin tavoin toimivan ja elementtejä luokittelevan neuroverkon, avulla voisi olla mahdollista muodostaa vierekkyyssrelaatiojärjestelmä, joka kuvaa kaikkea tietoa, mitä neuroverkko on tunnistanut. Tuloksena syntyisi vierekkyyssrelaatiojärjestelmä johon lisättävä tieto kuvaisi eräänlaista tunnistetun maailman skeemaa. Toisaalta neuroverkko voisi luokitella kaiken tunnistamansa tiedon, jolloin vierekkyyssrelaatiojärjestelmä kuvaisi 'kaikkea tietoa'.

Neuroverkoille opetetaan yksi tehtävä, jolloin niitä niiden voidaan ajatella olevan erikoistuneita hermoston osien tavoin aivoissa. Jos neuroverkko saa tiedot syötekerrokseensa toiselta neuroverkolta, ja edelleen antaa seuraavalle neuroverkolle käsittelemänsä tiedon, voidaan muodostaa pidempi useista erikoistuneista neuroverkoista muodostettu neuroverkko. Yhdistettyjen neuroverkkojen neuroverkon edellytyksenä olisi siten, että

neuroverkot tulisi ylipäänsä pystyä yhdistämään toisiinsa. Yhdistäminen tapahtuisi mahdollisesti syöte- ja tulostekerroksen neuroneista, joista syötekerroksen neuronit eivät prosessoivat tietoa ja niihin liittyy ainoastaan lähteviä kaaria. Tulostekerroksen neuronit taas prosessoivat tietoa ja niihin liittyy ainoastaan tulevia kaaria. Edellytyksiä yhdistämiselle vaikuttaa olevan. Vierekkyyssrelaatiojärjestelmään neuroverkkojen yhdistäminen ei tuo uusia tyyppejä, ainoastaan uusia elementtejä ja uusia relaatioita. Yhdistettyjen neuroverkkojen neuroverkon alku eli järjestyksessä ensimmäisen neuroverkon syötekerroksen neuronit voivat ottaa tiedon esimerkiksi fyysisestä maailmasta sensorien avulla tai ohjelmallisesti esimerkiksi tietoliikenteestä aistien tavoin. Neuroverkon loppu voi olla esimerkiksi tulostin, puhesyntetisaattori tai mahdollisesti mikä tahansa ohjelmointirajapinta. Voisi olla täysin mahdollista että syöte- ja tulostekerroksia olisi useita ja eri vaiheissa neuroverkkoa, jolloin ne voisivat jakaa osan erikoistuneista neuroverkoista.

Eräänlaisen itseohjautuvan oppimisen edellytyksenä voisi pitää oman toimintansa ja siihen liittyvien syy-seuraus -suhteiden tunnistamista. Järjestelmän oman toimintansa seurausten tunnistamisen edellytyksenä on takaisinkytkentä ja takaisinkytkennän kautta saadun tiedon tunnistaminen – takaisinkytkennän tunnistaminen.

6 JOHTOPÄÄTÖKSET

Neuroverkon rakenne ja rakenteen ominaisuudet voitiin kuvata viereisyysrelaatiojärjestelmällä. Kuvausta varten neuroverkko määriteltiin koostuvan kolmesta komponentista ja kahdesta muuttuvasta ominaisuudesta. Neuroverkon komponentit ja ominaisuudet määriteltiin omiksi tyypeikseen, jolloin vierekkyyssrelaatiojärjestelmään saatiin viisi tyyppiä. Relaatiot määriteltiin mahdollisimman yksinkertaisesti neuroverkon osien riippuvuussuhteiden perusteella. Esimerkiksi kaaren painon olemassaolo on riippuvainen kaaren olemassaolosta, joten vain näiden kahden osan välillä on relaatio. Kaari taas ei ole suoraan riippuvainen kerroksesta, vaan neuronista. Neuronista taas kuuluu johonkin kerrokseen ja on siten riippuvainen kerroksesta.

Vierekkyyssrelaatiojärjestelmässä ei ole aikakomponenttia, mutta kuvauksen voidaan ajatella esittävän kohteen rakennetta tietyllä ajan hetkellä. Neuroverkon rakenteen muuttumista voidaan kuvata muodostamalla useita kuvauksia neuroverkosta. Muutoksen kuvaamista varten on oltava olemassa komponentti, joka kertoo kuvausten järjestyksen ja ilmaisee siten ajallisuutta. Kuvauksen kohteen sisäistä tilaa voi olla mahdollista esittää ominaisuuksien arvojen avulla, mutta esimerkiksi neuroverkon toimintaa ei pystytä kuvaamaan. Arvojen esittämistä varten olisi joissain tapauksissa oltava kuvattavissa ja esitettävissä vierekkyyssrelaatiojärjestelmällä ääretön joukko numeerisia arvoja.

Tässä työssä toteutettu neuroverkon kuvaus edustaa kaikkia esiteltyjä neuroverkkoja siten, että sen tyyppiä voidaan käyttää myös muiden esiteltyjen neuroverkkojen kuvaamiseen. Toisin sanoen, määritelty kuvaus on siten yleinen, että se kuvaa kaikkia neuroverkkoja. Kuvattavat neuroverkot voivat olla vaihtelevan kokoisia tai ne voivat olla erityyppisiä, eli niissä voi olla vaihteleva määrä elementtejä, mutta ei vaihtelevaa määrää tyyppiä.

Vierekkyyssrelaatiojärjestelmällä kuvattu neuroverkko voidaan käydä algoritmisesti läpi siihen sisältyvän verkon kautta. Toteutuksessa se voi tarkoittaa esimerkiksi vierekkyyssrelaatiojärjestelmän sisältämän tiedon kirjoittamista tallennettavaan muotoon tai lukemista tallennustilasta käsiteltävään muotoon. Tietotyypin avulla voidaan toteuttaa esimerkiksi ohjelmointikirjasto kirjoittamalla tietotyyppille tietorakenteille tyyppilliset käsittelyt.

Tätä työtä seuraavia jatkotutkimuksen kohteita voisivat olla esimerkiksi vierekkyyssrelaatiojärjestelmässä syntyvien muutosten tutkiminen ja hallinta sekä tiedon ”irroittaminen” neuroverkosta vierekkyyssrelaatiojärjestelmän avulla. Muutos saattaa tehdä vierekkyyssrelaatiojärjestelmästä käyttökelvottoman, jolloin se ja mahdollisesti relaatiokombinaatio pitää määritellä uudestaan. Jos uudelleenmäärittely voi tehdä algoritmisesti, voidaan se todennäköisesti tehdä myös automaattisesti. Vierekkyyssrelaatiojärjestelmästä tulisi tällöin itsensä korjaava järjestelmä. Uusien elementtien ja tyyppien luokittelussa voisi olla mahdollista käyttää neuroverkkoa.

Vierekkyyssrelaatiojärjestelmän hyötyjä tiedon irroittamisessa neuroverkosta eli neuroverkon sisäisen tilan tulkinnassa tulisi tutkia. Kuten myös vierekkyyssrelaatiojärjestelmän käyttämistä järjestelmien mallinnuksen välineenä. Lisäksi vierekkyyssrelaatiojärjestelmän liittyvä elementtien yhdistäminen tilan käytön tai läpikäynnin optimoimiseksi, kuten myös yhdistymättömyys esimerkiksi molekyylien reagoimisen tutkimisessa. Lopuksi, vierekkyyssrelaatiojärjestelmän tietotyypille voitaisiin kirjoittaa tarpeelliset käsittelyt, esimerkiksi elementtien lisäämiselle ja poistamiselle, jolloin tietotyyppi ja käsitte-lyt yhdessä voisivat muodostaa tietorakenteen.

LÄHTEET

- Bishop J.M. & R.J. Mitchell (1991). Neural networks - an introduction. *Neural Networks for Systems: Principles and Applications, IEE Colloquium on* [pdf] [16.12.2015], Vaasa. Saatavissa: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=180904&contentType=Conference+Publications>
- Bose N. K. & P. Liang (1996). *Neural Network Fundamentals with Graphs, Algorithms, and Applications*. New York: McGraw-Hill. ISBN: 0-07-006618-3.
- CC = Creative Commons [8.3.2016], <https://commons.wikimedia.org/wiki/File:Hopfield-net-vector.svg>
- Fowler, Martin (2005). *UML Distilled. A Brief Guide to the Standard Object Modeling Language*. 3. painos. Boston: Pearson Education, Inc. ISBN: 0-321-19368-7.
- Hertz, John, Anders Krogh & Richard G. Palmer (1993). Introduction to the Theory of Neural Computation. A Lecture Notes Volume in the Santa Fe Institute in the Science of Complexity. Redwood City: Addison-Wesley Publishing Company. ISBN: 0-201-51560-1.
- Hrbacek, Karel & Thomas Jech (1999). Introduction to Set Theory. New York: Marcel Dekker, Inc. 310 sivua. ISBN: 0824779150.
- Kokkarinen, Ilkka & Kirsti Ala-Mutka (2002). Tietorakenteet ja algoritmit. Jyväskylän: Gummerus Kirjapaino Oy. 421 sivua. ISBN: 951-762-795-5.
- McCulloch, Warren & Walter Pitts (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*. 5: -. Sivut 115-133.
- Mäenpää, Teemu (2015). *Utilization of adjacency model in graph analysis*. Vaasan yliopisto. Teknillinen tiedekunta. Akateeminen väitöskirja. ISBN: 978-952-476-643-2.

- Mäenpää, Teemu & Vesa Nyrhilä (2013a). Framework for representing Semantic Link Network with Adjacency Relation System. *Proceedings of the 2nd International Conference on Integrated Information*. Procedia - Social and Behavioral Sciences 73. Sivut 438-443.
- Mäenpää, Teemu & Vesa Nyrhilä (2013b). Visualizing and Structuring Semantic Data. *International Journal of Machine Learning and Computing*. 3: 2. Sivut 209-213.
- Mäenpää, Teemu & Merja Wanne (2015). Review of Similarities between Adjacency Model and Relational Model. *Modelling, Computation and Optimization in Information Systems and Management Sciences*. Advances in Intelligent Systems and Computing 360. Sivut 69-79.
- Svozil, Daniel, Vladimir Kvasnicka & Jiri Pospichal (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems* [pdf] 39: 1 [17.12.2015], Vaasa. Saatavissa: <http://www.sciencedirect.com/science/article/pii/S0169743997000610>
- Töyli, Jari (2002). *Modeling Semistructured Data by the Adjacency Model*. Vaasan yliopisto. Teknillinen tiedekunta. Lisensiaatin tutkielma. Vaasa.
- Töyli, Jari (2006). *AdSchema – a Schema for Semistructured Data*. Vaasan yliopisto. Teknillinen tiedekunta. Akateeminen väitöskirja. ISBN: 952-476-131-9.
- Uhrig R.E. (1995). Introduction to artificial neural networks. Industrial Electronics, Control, and Instrumentation, 1995., Proceedings of the 1995 IEEE IECON 21st International Conference on [pdf] 1: [16.12.2015], Vaasa. Saatavissa: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=483329&newsearch=true&queryText=10.1109%2FIECON.1995.483329>
- UML 2.4.1 = UML Spesifikaatio versio 2.4.1 (2011). Object Management Group. Saatavissa: <http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF>
- Wanne, Merja (1998). *Adjacency Relation Systems*. Vaasan yliopisto. Teknillinen tiedekunta. Akateeminen väitöskirja. ISBN: 951-683-703-4.

Voloshin, Vitaly I. (2009). Introduction to Graph Theory. New York: Nova. 160 sivua.
ISBN: 9781614701132.

Zou, Jinming, Yi Han & Sung-Sau So (2008). Overview of Artificial Neural Networks.
Teoksessa: *Artificial Neural Networks: Methods and Protocols*, 15-23. Ed. D. S.
Livingstone. NJ 07512 USA: Humana Press. ISBN: 978-1-61737-738-9.