

**Neuroverkoilla luokittelu ja tapausten keinotekoinen lisääminen
aineistoon**

Lassi Autio

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos
Tietojenkäsittelyoppi
Pro gradu -tutkielma
Elokuu 2003

Tampereen yliopisto
Tietojenkäsittelytieteiden laitos
Tietojenkäsittelyoppi
Lassi Autio: Neuroverkoilla luokittelu ja tapausten keinotekoinen lisääminen
aineistoon
Pro gradu -tutkielma, 88 sivua
Elokuu 2003

Tutkielmassa tarkastellaan neuroverkkojen kykyä luokitella erilaisia lääketieteellisiä aineistoja ja sitä miten tapausten keinotekoinen lisääminen aineistoon vaikuttaa neuroverkkojen kykyyn luokitella. Neuroverkoille opetettiin kahdeksan eri aineistoa käyttäen kymmenkertaista ristiinvalidointia. Jokaiselle aineistolle etsittiin sopiva neuroverkon arkkitehtuuri ja opetusparametrit. Jokaisen aineiston kohdalla neuroverkot oppivat parhaiten suurimman luokan. Opetusten tulokset havainnollistivat myös sitä miten monta piiloneuronia neuroverkossa tulee olla luokkien lukumäärän suhteen.

Kolmeen kahdeksasta aineistosta lisättiin keinotekoisesti tapauksia pienimpiin luokkiin. Tällä tavalla saatiin suurennettua aineistoa ja vähennettyä luokkajakauman vinoutta. Tapausten keinotekoinen lisääminen ei vaikuttanut paljoa neuroverkkojen luokittelutarkkuuteen, mutta paransi usein pienempien luokkien luokittelua ja huononsi suurimman luokan luokittelua.

Avainsanat ja -sanonnat: neuroverkot, luokitleminen, koneoppiminen.

Sisällys

1. Johdanto.....	1
2. Neuroverkot.....	3
2.1. Johdanto neuroverkkoihin	3
2.2. Neuroverkkojen rakenne	4
2.3. Perceptron.....	6
2.4. Monikerroksinen perceptron	8
2.5. Opettaminen.....	12
2.5.1. Suunnattu opettaminen	13
2.5.2. Suuntaamaton opettaminen ja SOM	20
2.5.3. Muita opetustapoja	22
2.5.4. Neuroverkon ylioppiminen ja arkkitehtuuri.....	22
2.6. Muita neuroverkkoja.....	26
2.6.1. Radiaaliantafunktioneuroverkko	26
2.6.2. Assosiatiivimuistit	26
2.6.3. Hopfieldin-verkko ja Boltzmannin kone.....	27
2.6.4. Takaisinkytketyvät neuroverkot	28
3. Aineistot.....	29
3.1. Aineistojen esittely.....	30
3.2. Aineistojen muokkaaminen neuroverkkoja varten	32
4. Neuroverkkojen muodostaminen aineistoille	36
4.1. Alokas.....	38
4.2. Buba	39
4.3. Ecoli.....	42
4.4. Haberman	46
4.5. Huimaus.....	48
4.6. New-thyroid	55
4.7. Pima	60
4.8. Virtsainkontinenssi.....	62
5. Tapausten keinotekoinen lisääminen.....	70
5.1. Tapausten keinotekoinen lisääminen huimaus-aineistoon	72
5.2. Tapausten keinotekoinen lisääminen Pima-aineistoon.....	75
5.3. Tapausten keinotekoinen lisääminen virtsainkontinenssi-aineistoon	78
5.4. Yhteenveto tapausten keinotekoisesta lisäämisestä	82
6. Yhteenveto.....	83
Viiteluettelo.....	86

1. Johdanto

Tässä tutkielmassa käsitellään neuroverkkojen teoriaa ja neuroverkkoja luokittelijoina kahdeksalla eri aineistolla. Lisäksi on tutkittu miten tapausten keinotekoinen lisääminen aineistoon vaikuttaa neuroverkkojen oppimiseen alkuperäiseen aineistoon verrattuna.

Neuroverkot ovat *koneoppimismenetelmä* (machine learning method), jotka pyrkivät mallintamaan aivojen toimintaa matemaattisesti. Koneoppimismenetelmät ovat tekoälyn osa-alue, joka tutkii laskennallisia menetelmiä, jotka parantavat suorituskykyään oppimalla [Viikki, 2002].

Suurin osa tässä tutkielmassa käytetyistä kahdeksasta aineistosta on lääketieteellisiä. Neuroverkkoja on käytetty useissa lääketieteellisissä sovelluksissa [Guerriere and Detsky, 1991] ja se oli ensimmäisiä sovellusalueita, johon neuroverkkoja alettiin käyttää [Baxt, 1995]. Neuroverkot pystyvät tekemään vaikeitakin diagnooseja, mihin niiden käyttökelpoisuus lääketieteessä ja muualla perustuu [Maclin and Dempsey, 1992]. Neuroverkkoja on käytetty lääketieteessä mm. syövän diagnosointiin [Maclin *et al.*, 1991; Maclin and Dempsey, 1992], sydänsairauksien diagnosointiin [Baxt, 1995; Kennedy *et al.*, 1997] ja kuvien ja laboratorionäytteiden analysointiin [Astion and Wilding, 1992]. Neuroverkoilla pystytään antamaan diagnoosi heti paikanpäällä ilman pitkiä tutkimuksia, mikä nopeuttaa diagnosointia [Wu and Yen, 1992]. Esimerkiksi Baxt [1995] kertoo tilanteen, missä neuroverkot pystyivät heti toteamaan potilaalla olleen akuutti sydänsairaus, kun vastaavat laboratoriotulokset antoivat saman tuloksen vasta neljän tunnin kuluttua. Akuutin sydänsairauden diagnosoiminen sairaalassa on erityisen vaikeaa, mutta opetetut neuroverkot ovat pystyneet tekemään diagnoosin nopeammassa ajassa [Kennedy *et al.*, 1997]. Neuroverkot olivat myös parempia akuutin sydänsairauden diagnosoissa: neuroverkko antoi 97 prosentin tarkkuudella oikean diagnoosin, kun lääkärin antaman diagnoosin tarkkuus oli 78 prosenttia [Baxt, 1991].

Neuroverkot pystyvät antamaan jopa suurilla todennäköisyyksillä oikeita tuloksia saamistaan syötteistä, kun niitä on opetettu. Usein neuroverkot antavat parempia tuloksia kuin lääkärit ja muut asiantuntijat [Baxt, 1995]. Neuroverkkojen käyttäminen saattaa olla joskus ongelmallista, koska ne eivät kerro miksi ovat päätyneet antamiinsa tuloksiin [Kennedy *et al.*, 1997]. Esimerkiksi lääketieteellisissä sovelluksissa lääkäreiden ja hoitajien on luotettava neuroverkon antamiin tuloksiin, vaikka he eivät tietäisi tarkkaan miksi neuroverkko on antanut kyseisen tuloksen. Neuroverkko on *musta*

laatikko (black box), jonka sisäistä toimintaa ei välttämättä pystytä tietämään [Guerriere and Detsky, 1991].

Tutkielman tavoitteena oli opettaa erilaisia lääketieteellisiä aineistoja neuroverkoille ja tarkastella miten neuroverkot pystyvät luokittelemaan niitä. *Luokittelussa* (classification) ennustetaan kategorisia *luokkaleimoja* (class label) [Viikki, 2003]. Monia tutkielmassa käytetyistä aineistoista on jo aikaisemmin luokiteltu Tampereen yliopiston tietojenkäsittelytieteiden laitoksella erilaisilla koneoppimismenetelmillä. Tutkielmassa on opetettu näitä aineistoja neuroverkoille. Lisäksi on kokeiltu lisätä kolmeen aineistoon keinotekoisesti tapauksia Swinglerin [1996] esittämällä tavalla. Tällä tavalla luokkajakauman vinoutta on saatu vähennettyä ja aineistojen kokoja suurennettua, mikä teoriassa antaa paremmat mahdollisuudet aineiston oppimiseen.

Luvussa 2 tarkastellaan neuroverkkojen teoriaa keskittyen perceptron-neuroverkkoihin. Aluksi selitetään mitä neuroverkot ovat ja minkälaisia niiden rakenteet ovat. Sen jälkeen syvennyttään neuroverkkojen opettamiseen painottuen suunnattuun opettamiseen. Luvun lopussa esitellään myös muutamia muita neuroverkkoja.

Luvussa 3 esitellään kaikki kahdeksan tutkielmassa käytettyä aineistoa. Lisäksi luvussa käsitellään keskeisiä menetelmiä, joilla aineistot tulisi esikäsitellä ennen kuin niitä aletaan opettaa neuroverkoille.

Luku 4 käsittelee aineistoista saatuja tuloksia, kun ne on opetettu neuroverkoille. Jokainen aineisto on käsitelty erikseen ja niille on yritetty löytää neuroverkon arkkitehtuuri, jolla luokitseminen onnistuisi mahdollisimman hyvin. Kaikkien kahdeksan aineiston tuloksia on käsitelty erikseen yksityiskohtaisesti.

Luvussa 5 tutkitaan miten tapausten keinotekoinen lisääminen aineistoihin vaikuttaa neuroverkkojen oppimiseen. Neuroverkkoja on opetettu kolmella eri aineistolla, joihin on lisätty keinotekoisesti uusia tapauksia. Keinotekoisesti suurennetuilla aineistoilla opettettujen neuroverkkojen tuloksia on tarkasteltu luvussa tarkemmin.

Luvussa 6 tehdään yhteenveto tässä tutkielmassa saaduista tuloksista. Jokaisen aineiston opettamisesta neuroverkoille annetaan lyhyt yhteenveto. Luvun lopussa tiivistetään tapausten keinotekoisien lisäämisen vaikutus neuroverkkojen oppimiseen.

2. Neuroverkot

2.1. Johdanto neuroverkkoihin

Keinotekoiset neuroverkot (artificially neural networks, neural networks) (jatkossa neuroverkot) ovat biologisten neuroverkkojen matemaattinen malli. Esimerkiksi ihmisen aivot ovat eräs biologinen neuroverkko. Neuroverkot oppivat aivojen tavoin erilaisia asioita ja osaavat opettamisen jälkeen laskea saamastaan syötteestä tuloksen. Esimerkiksi jos neuroverkko on opetettu tunnistamaan a-kirjaimia (kuva 1), niin se osaa kertoa, minkä kirjaimen se on saanut syötteenä. Tämä vastaa ihmistä joka nähdessään kirjaimen tunnistaa, mikä kirjain se on. Neuroverkot eivät kuitenkaan ole täydellinen kopio aivoista tai muista biologisista neuroverkoista, vaan niiden lähtökohta, toimintaperiaate ja -tavat ovat samankaltaisia kuin biologisilla neuroverkoilla. Neuroverkkojen ja biologisten neuroverkkojen yhteisiä piirteitä ovat mm.

- oppiminen virheen ja onnistumisen kautta
- yleistäminen
- vikasietoisuus.

Neuroverkot eivät aluksi osaa tehdä mitään järkevää. Neuroverkko on opetettava, jotta se osaisi tehdä jotain järkevää. Sama pätee ihmisellekin. Syntyessään ihminen ei osaa tehdä juuri mitään, mutta ajan myötä hän oppii esimerkiksi kävelemään. Opettamisen jälkeen neuroverkko pystyy toimimaan älykkäästi. Neuroverkot oppivat samalla tavalla kuin ihmisetkin, kun ne onnistuvat, niin ne vahvistavat onnistunutta suoritusta, mutta epäonnistuessa ne yrittävät korjata virhettään, jotta onnistuisivat seuraavalla kerralla. Esimerkiksi tietokoneen hiiren kaksoisklikkaus opitaan vähitellen. Erehdysten ja onnistumisten kautta ihminen oppii tekemään kaksoisklikkauksen, vaikka ensimmäisellä hiiren käyttökerralla se on vaikeaa.

Ihminen osaa hyvin lukea erilaisia käsialoja ja esim. erottaa mitä erilaisimpia a-kirjaimia (kuva 1), eli ihminen osaa yleistää asioita. Tietokoneelle erilaisten kirjainten tunnistaminen on vaikeaa, vaikka tietokone osaakin tehdä ihmistä nopeammin laskutoimituksia. Syynä tähän on, että ihmisen aivoissa toimii tuhansia soluja samanaikaisesti rinnakkain. Tehtävä hajautetaan moneen pieneen osaan eri soluille ja lopuksi nämä ratkaisut yhdistetään. Hajautuksen ja rinnakkaisuuden ansiosta aivot pystyvät tekemään tietokoneille monimutkaisia toimenpiteitä lyhyessä ajassa, kun taas tietokoneella on usein vain yksi tai kaksi prosessoria. Neuroverkon voi opettaa tunnistamaan kirjaimia, jonka jälkeen se osaa tunnistaa kirjaimen, jota se ei ole ennen nähnyt, kuten ihminenkin. Esimerkiksi ihminen pystyy lukemaan käsialaa, jota hän ei ole aikaisemmin

lukenut. Esimerkiksi ihminen erottaa kuvasta 1 hyvin kolme erilaista a-kirjainta.

A a Δ

Kuva 1. Kolme erilaista a-kirjainta. Neuroverkko voidaan opettaa tunnistamaan, että jokainen näistä kirjaimista on a.

Ihminen saattaa menettää osittain puhekykynsä aivovamman kautta. Nykyään on mahdollista saada puhekykyä tästä huolimatta takaisin. Neuroverkotkin pystyvät usein toipumaan samalla tavalla vioistaan. Tällöin esimerkiksi muut kunnossa olevat osat ottavat tehtäväkseen voittuneiden osien tehtävät tai opettavat voittuneet osat uudelleen. [Juhola, 2000]

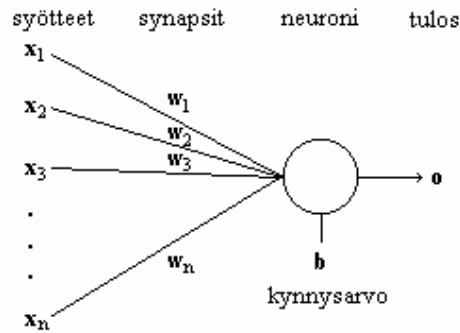
2.2. Neuroverkkojen rakenne

Neuroverkkoja on useita erilaisia, mutta kaikille niille yhteisiä piirteitä ovat [Callan, 1999]:

- niissä on joukko laskentayksiköitä,
- niissä on joukko laskentayksiköitä yhdistäviä kaaria,
- niissä on sääntö signaalien välittämiseksi ympäri verkkoa,
- niissä on sääntö yhdistää syötesignaalit,
- niissä on sääntö laskea tulossignaali,
- niissä on oppimisalgoritmi painoarvojen muokkaamista varten.

Neuroverkko koostuu useista laskentayksiköistä, eli *neuroneista* (neuron, unit) ja *synapsit* (synapse, edge) yhdistävät niitä toisiinsa. Neuronit saavat syötteinä eri arvoja muilta neuroneilta. Näistä arvoista lasketaan neuronin tulossignaali, jonka se lähettää seuraavalle neuronille, jos se ylittää neuronin sisäisen *kynnysarvon* (bias). Kynnysarvoa merkitään usein θ -kirjaimella. Neuronia ja siihen tulevia synapsia on havainnollistettu kuvassa 2.

Neuroni lähettää laskemansa tuloksen signaalina eteenpäin seuraavalle neuronille synapseja pitkin, joilla on jokin painoarvo (reaaliluku). Synapsin painoarvo voi joko vahvistaa samaansa signaalia tai heikentää sitä [Callan, 1999]. Neuroverkkojen älykkyys on tallennettuna synapsien painoarvoihin ja neuronien kynnysarvoihin.



Kuva 2. Neuroni ja siihen tulevat synapsit.

Yleensä neuroni yhdistää syötesignaalin seuraavan kaavan mukaan:

$$net = \sum_{i=1}^n w_i x_i - \theta, \quad (2.1)$$

missä net on laskettu arvo. Toinen tapa on liittää summalausekkeeseen kynnysarvo, jolloin kaava on muotoa

$$net = \sum_{i=0}^n w_i x_i, \text{ missä } x_0 = 1 \text{ ja } w_0 = -\theta. \quad (2.2)$$

Molemmat kaavat antavat saman tuloksen. Kukin syöte kerrotaan synapsin painolla, jolloin synapsin paino määrää syöteen tärkeyden neuronille. Syötesignaalin ja synapsien painoarvosta vielä vähennetään neuronin kynnysarvo. Neuronin lähettämä tulos lasketaan antamalla edellä laskettu net aktivaatiofunktiolle. Aktivaatiofunktio on sääntö laskea neuronin tulos. Aktivaatiofunktioiden antamat arvot ovat yleensä diskreettejä väleiltä $\{0, 1\}$ tai $\{-1, 1\}$ tai ne voivat antaa jatkuvia arvoja väleiltä $[0, 1]$ tai $[-1, 1]$. Aktivaatiofunktion tulee olla jatkuva ja se ei saisi olla lineaarinen [Swingler, 1996]. Esimerkki diskreetistä binäärisestä aktivaatiofunktioista on *Heavisiden funktio* (kuva 3)

$$f(net) = \begin{cases} 1, & \text{jos } net \geq \theta \\ 0, & \text{jos } net < \theta \end{cases} \quad (2.3)$$

ja jatkuvasta binäärisestä aktivaatiofunktioista *logaritminen*

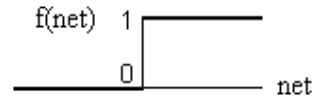
$$f_{\log}(net) = \frac{1}{1 + e^{-net}} \quad (2.4)$$

tai *tangenttinen sigmoidi funktio*

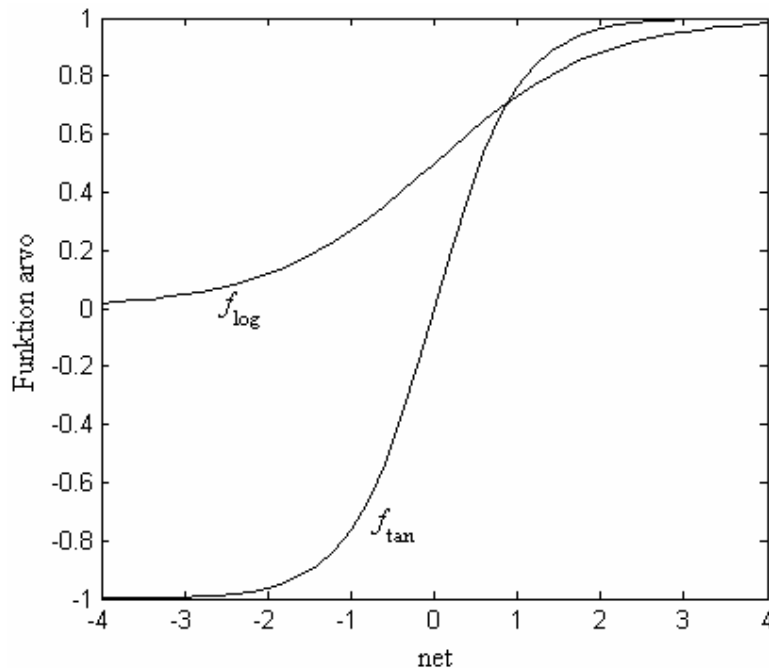
$$f_{\tan}(net) = \frac{2}{1 + e^{-2net}} - 1 \quad (2.5)$$

(kuva 4). Jatkuvilla aktivaatiofunktioilla, kuten sigmoidi funktiolla, pystytään laskemaan tarkempia arvoja kuin diskreeteillä aktivaatiofunktioilla, kuten Heavisiden funktiolla. Jatkuvilla arvoilla neuroverkko pystyy antamaan

äärettömän monta tulosta (esim. sigmoidi funktio), kun taas diskreeteillä arvoilla vain rajallisen määrän (esim. Heavisiden funktio). Sigmoidi funktiosta pystytään saamaan kynnsfunktion tavoin käyttäytyvä, jos ko. neuronin kynnsarvo ja synapsien painoarvot ovat suuria [Bishop, 1995].



Kuva 3. Heavisiden funktion kuvaaja [Callan, 1999].



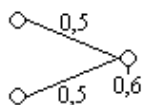
Kuva 4. Logaritminen sigmoidi funktio (f_{\log}) kuvaaja [Callan, 1999] ja tangenttinen sigmoidi funktio (f_{\tan}) kuvaaja [MATLAB, 2002].

Yhdessä neuroverkossa on aina vähintään yksi *syöteneuroni* (input neuron) ja *tulosneuroni* (output neuron). Syöteneuronit eivät laske mitään, vaan ne ottavat vastaan syötteitä ja välittävät ne eteenpäin eikä niillä ole kynnsarvoa. Syöteneuroneista koostuu neuroverkon *syötekerros* (input layer) ja tulosneuroneista *tuloskerros* (output layer). Joissakin neuroverkoissa syöte- ja tuloskerros ovat samoja, kuten *Hopfieldin verkossa* (Hopfield network). Syöteneuronit saavat syötesignaalin, joka kulkee synapseja pitkin tulosneuroneille, jotka antavat verkon tuloksen.

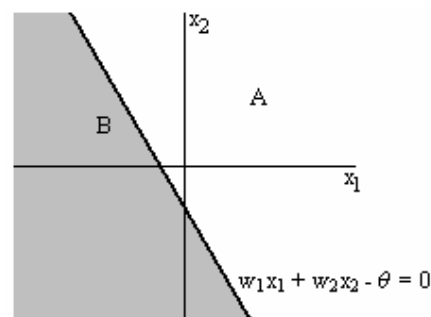
2.3. Perceptron

Yksinkertaista neuroverkkoa, jossa on syöteneuroneja ja yksi tulosneuroni kutsutaan *perceptroniksi* (perceptron, linear perceptron) tai yksikerroksiseksi

perceptroniksi. Kuvan 5 perceptron antaa tuloksena loogisen AND-funktion kun käytetään Heavisiden funktiota aktivaatiofunktiona, mikä on yleisin aktivaatiofunktio perceptroneissa. Oletetaan, että verkko saa syötteen (1, 0). Tällöin tulosneuronin *net* lasketaan kaavan (2.1) tai (2.2) mukaan, jolloin tulokseksi saadaan -0,1. Muuttujan *net* arvo annetaan nyt Heavisiden funktiolle, jolloin neuroverkko antaa tuloksena 0. Vastaavasti syötteen ollessa (1, 1) tulosneuronin *net* on 0,4 ja neuroverkon tulos on 1. Yleistä perceptronin mallia voidaan havainnollistaa geometrisesti koordinaatistossa, jossa suora jakaa kaksi luokkaa (kuva 6).



Kuva 5. Loogisen AND-funktion laskeva perceptron.

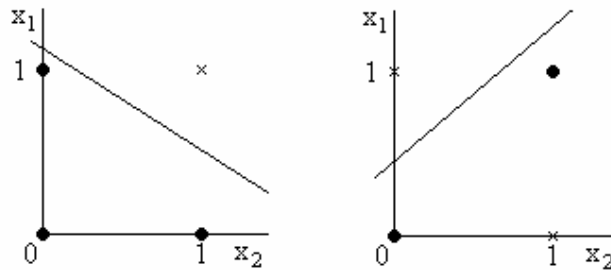


Kuva 6. Suora $w_1x_1 + w_2x_2 - \theta = 0$ jakaa luokat A ja B erilleen, missä x_1 ja x_2 ovat syöte, w_1 ja w_2 ovat synapsien painoarvot ja θ on kynnsarvo. [Haykin, 1994]

Perceptronin vetää hahmoavaruuteen suoran tai tason, jonka avulla se pystyy erottamaan kaksi lineaarisesti erottuvaa luokkaa [Juhola, 2000]. Perceptronin heikkoutena on, että se pystyy ratkaisemaan vain lineaarisia ongelmia [Picton, 2000]. Esimerkiksi perceptron ei pysty ratkaisemaan loogista XOR-funktiota (kuva 7), koska sitä ei voida ratkaista lineaarisesti. Ongelman ollessa kaksi- tai kolmiulotteinen (so. syöteneuroneja on kaksi tai kolme), voidaan lineaarisista eroavuutta havainnollistaa geometrisesti joko suoralla (kaksiulotteinen) tai tasolla (kolmiulotteinen). Geometrista havainnollistusta ei voida kuitenkaan tehdä, jos ongelma on n -ulotteinen. Rojas [1996] määrittelee lineaarisen eroavuuden n -ulotteisessa avaruudessa:

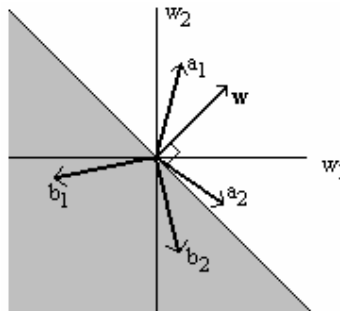
"Kahta joukkoa pisteitä A ja B n -ulotteisessa avaruudessa sanotaan lineaarisesti erottuviksi, jos on olemassa $n+1$ reaalilukua w_1, w_2, \dots, w_{n+1} , että jokainen piste $(x_1, x_2, \dots, x_n) \in A$ toteuttaa epäyhtälön $\sum w_i x_i$

$\geq w_{n+1}$ ja jokainen piste $(x_1, x_2, \dots, x_n) \in B$ toteuttaa epäyhtälön $\sum w_i x_i < w_{n+1}$.



Kuva 7. Vasemmalla oleva looginen AND-funktio pystytään ratkaisemaan lineaarisesti, mutta oikealla olevaa loogista XOR-funktiota ei pystytä.

Perceptron voidaan havainnollistaa geometrisesti myös vektorinäkökulmasta. Perceptronin vektorinäkökulmassa jokainen tapaus on vektori $\mathbf{x} = (x_1, x_2, \dots, x_n)$ n -ulotteisessa avaruudessa, missä n on syötöneuronien lukumäärä. Neuroverkon painoarvot määritellään tällöin myös vektorina $\mathbf{w} = (w_1, w_2, \dots, w_n)$, missä n syötöneuronien lukumäärä. Tässä tapauksessa tulosneuronin kynnyisarvo on 0. Oletetaan että on kaksi luokkaa $A \in \mathbb{R}^2$ ja $B \in \mathbb{R}^2$ ja molemmilla luokilla kaksi vektoria. Nyt painoarvovektori $\mathbf{w} = (w_1, w_2)$ on normaali sille suoralle, joka jakaa luokat A ja B (kuva 8). Tällöin luokan A vektorien pistetulo painoarvovektorin \mathbf{w} kanssa on suurempi kuin 0 ja vastaavasti luokan B tapauksessa pienempi kuin 0. [Rojas, 1996]



Kuva 8. Yksikerroksisen perceptronin geometrinen visualisointi vektorinäkökulmasta. [Rojas, 1996]

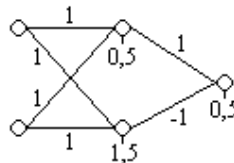
2.4. Monikerroksinen perceptron

Monikerroksinen perceptron (multi layer perceptron) on monimutkaisempi neuroverkko kuin perceptron. Monikerroksisessa perceptronissa on myös *piiloneuroneja* (hidden neuron), jotka muodostavat yhden tai useamman *piilokerroksen* (hidden layer). Piilokerrokset sijaitsevat yleensä syöte- ja

tuloskerroksen välissä. Monikerroksisella perceptronilla pystytään ratkaisemaan monimutkaisempia ongelmia kuin yksikerroksisella perceptronilla.

Neuroverkkojen yhteydessä on erilaisia käytäntöjä ilmaista, kuinka monta kerrosta neuroverkossa on. Usein lasketaan kerroksiksi vain ne kerrokset, joihin liittyy synapsit painokertoimineen. Tällöin syötekerrosta ei lasketa kerrokseksi, koska siihen ei liity synapseja. Tällä laskutavalla kuvan 9 neuroverkko olisi kaksikerroksinen perceptron. Toinen tapa laskea neuroverkon kerrosten määrä on laskea myös syötekerros yhdeksi kerrokseksi. Tällä tavalla laskettuna kuvan 9 neuroverkko on kolmikerroksinen perceptron. Tässä tutkielmassa käytetään jälkimmäistä tapaa laskea neuroverkon kerrosten lukumäärä, jossa lasketaan neuroverkon kaikki kerrokset.

Aikaisemmin todettiin, että loogista XOR-funktiota (joko tai) ei pystytä ratkaisemaan yksikerroksisella perceptronilla. Silti esimerkiksi kuvan 9 monikerroksinen perceptron pystyy ratkaisemaan sen. Kun syötteeksi annetaan (0, 1) lasketaan aluksi piilokerroksen tulokset. Piiloneuronien *net*-arvoiksi kaavalla (2.2) saadaan 0,5 ja -0,5, joista edelleen Heavisiden funktion jälkeen 1 ja 0 piiloneuronien tuloksiksi. Tulosneuronin laskee piiloneuronien tuloksista samalla tavalla oman *net*-arvonsa 0,5 ja antaa neuroverkon tuloksena Heavisiden funktion jälkeen 1. Vastaavasti syötteen ollessa (1, 1) neuroverkon tuloksena on 0.



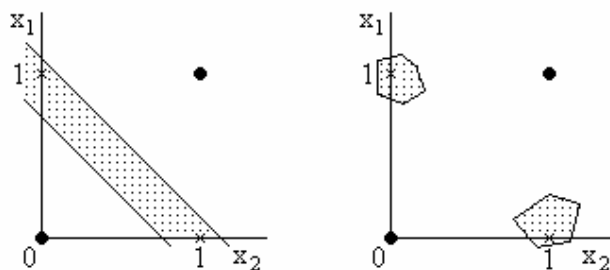
Kuva 9. Loogisen XOR-funktion laskeva monikerroksinen perceptron. Vasemmalta syöte-, piilo- ja tuloskerros. Ylempi piiloneuronin ratkaisee loogisen OR-funktion ja alempi piiloneuronin loogisen AND-funktion. Tulosneuronin yhdistää nämä tulokset ja antaa loogisen XOR-funktion. [Juhola, 2000].

Käydään vielä läpi tarkemmin, miten monikerroksinen perceptron laskee tuloksen. Monikerroksisessa perceptronissa jokainen piilo- ja tulosneuronin laskee saamistaan syötteistä *net*-arvon kaavalla 2.2, jonka jälkeen ko. neuronin tulos saadaan antamalla *net* aktivaatiofunktiolle. Kolmikerroksisen perceptronin tapauksessa tulosneuronin *k* tulos saadaan laskettua kaavalla

$$o_k = f_k(net_k) = f_k\left(\sum_{j=0}^m w_{jk} f_j(net_j)\right) = f_k\left(\sum_{j=0}^m w_{jk} f_j\left(\sum_{i=0}^n w_{ij} x_i\right)\right), \quad (2.6)$$

missä k on tulosneuronin indeksi, j on piiloneuronin indeksi, i on syötöneuronin indeksi ja m on piilo- ja n syötöneuronien määrä. Yhdellä piilokerroksella pystytään rajaamaan alueita piiloneuronien määrän suorien tai tasojen verran [Fu, 1994]. Looneyn [1997] mukaan suorien rajoittamien alueiden yläraja syötöneuronien lukumäärän ollessa i on 2^i ja alaraja on $i+1$. Kun neuroverkkoon liitetään toinen piilokerros, niin pystytään tuottamaan monimutkaisempia alueita (kuva 10). Monen suoran tulosten yhdistäminen kuvaa hyvin monikerroksisen perceptronin paremmuutta yksikerroksiseen verrattuna.

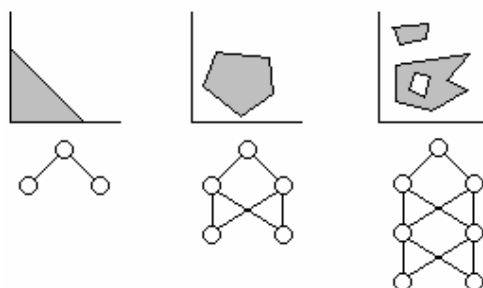
Kolmogorovin teoreeman (Kolmogorov's theorem) mukaan jatkuva $n:n$ muuttujan funktio voidaan aina esittää käyttäen äärellistä yhden muuttujan funktioiden yhdistelmää [Rojas, 1996]. Neuroverkkojen mielessä teoreema sanoo, että mikä tahansa funktio $f(\mathbf{x})$, missä $\mathbf{x} = (x_1, \dots, x_n)$, voidaan esittää kahden piilokerroksen neuroverkolla, missä ensimmäisessä piilokerroksessa on $n(2n+1)$ piiloneuronia ja toisessa piilokerroksessa $2n+1$ piiloneuronia. Teoreeman mukaan kaksi kerrosta piiloneuroneja riittää rajaamaan mielivaltaisia alueita. Teoreema sanoo vain, että tällainen neuroverkko on olemassa, mutta se ei kerro miten esim. painoarvot tulisi asettaa. [Bishop, 1995]



Kuva 10. Vasemmalla monikerroksinen perceptronin ratkaisu loogiseen XOR-funktioon yhdellä piilokerroksella, jossa on kaksi piiloneuronia. Oikealla on sama useammalla piilokerroksella ja -neuronilla.

Ensimmäinen piilokerros tekee hahmoavaruuteen siinä olevien piiloneuronien lukumäärän suorita. Ensimmäisen piilokerroksen jälkeen oleva kerros, joka on joko toinen piilokerros tai tuloskerros, muodostaa ensimmäisen kerroksen suorista kombinaatioita. Nämä kombinaatiot muodostavat *konvekseja alueita* tai *konvekseja peitteitä*. Konvekseksi peite on alue, jossa alueen mikä tahansa piste voidaan yhdistää mihin tahansa alueen toiseen pisteeseen janalla, joka ei ylitä alueen reunaviivaa. Alue on avoin, jos sen määrittää kaksi suoraa, kuten kuvan 10 vasen ratkaisu. Useamman kuin kahden suoran alue on suljettu, mikäli suorat ovat erisuuntaisia. Kuvan 10 oikealla olevassa ratkaisussa on

konveksit alueet, jotka ovat muodostuneet kuudesta ja viidestä suorasta. Toisen piilokerroksen jälkeinen kerros saa syötteenään toisen piilokerroksen konveksit alueet, joiden kombinaatiot eivät ole enää konvekseja. Tämä kerros pystyy rajaamaan mielivaltaisen monimutkaisia alueita ja luokittelemaan mitä luokkia tahansa. Tämän takia monikerroksisessa perceptronissa ei välttämättä tarvitse olla kuin kaksi piilokerrosta, mikä vahvistaa Kolmogorovin teoremaa. Piilokerrosten lukumäärän vaikutusta luokittelemisessa havainnollistaa kuva 11. Monikerroksisten perceptronien kyky tunnistaa moniulotteisia yhteyksiä lääketieteellisessä aineistossa mahdollistaa neuroverkoille hyvän suorituskyvyn lääketieteessä [Baxt, 1995]. [Juhola, 2000]



Kuva 11. Geometrinen esitys piilokerrosten määrän vaikutuksesta päätösalueeseen. [Bishop, 1995]

Piiloneuroneja ja -kerroksia voidaan ajatella matemaattisten funktioiden astelukuina. Toisen asteen yhtälöllä (x^2) pystytään määrittellä yksi käännöspiste, mutta kolmannen asteen yhtälöllä (x^3) pystytään määrittellä kaksi käännöspistettä [Swingler, 1996]. Samalla tavalla voidaan ajatella piilokerrosten ja -neuronien suhteen. Mitä enemmän neuroverkossa on piiloneuroneja ja -kerroksia, sitä monimutkaisempia aineistoja neuroverkko pystyy oppimaan.

Neuroverkon synapsien painoarvot kullekin kerrokselle havainnollistetaan usein matriiseina. Jokaisen kerroksen matriisissa kukin sarake merkitsee yhtä kerroksen neuronin ja rivi neuronin liittyvän synapsin painoarvoa siten että ensimmäisellä rivillä on $-\theta$ (-kynnysarvo). Neuroverkon syötekin voidaan havainnollistaa matriisina. Silloin kukin matriisin rivi on syöte ja sarakkeet ovat syöteneuroneja siten, että ensimmäisessä sarakkeessa on kynnysarvojen kertoimet, eli 1. Näillä matriiseilla pystytään myös laskemaan neuronien *net*-arvoja (kuva 12). Kun kerrotaan syöttestä tehty matriisi tuloskerroksen matriisilla yksikerroksisessa perceptronissa, saadaan tuloksena tulosneuronin *net*.

$$[1 \ 0 \ 1] \cdot \begin{bmatrix} -0.5 & -1.5 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} = [0.5 \ -0.5]$$

Kuva 12. Kuvan 9 neuroverkon syötteen (0, 1) ja piilokerroksen matriisiesitys ja näiden tulo, josta saadaan piiloneuroneiden *net*.

2.5. Opettaminen

Yleensä aluksi neuroverkon painokertoimet ovat pieniä satunnaislukuja ja neuroverkko ei anna järkeviä tuloksia. Neuroverkot eivät aluksi osaa mitään, vaan ne on opetettava. Neuroverkkojen opetuksessa yritetään etsiä parhaat painoarvot opetettavalle neuroverkolle. Opetuksessa neuroverkolle annetaan joukko syötteitä, joiden ominaisuudet verkko pyrkii oppimaan. Oppimiseen käytetään opetusalgoritmia. Opetusalgoritmeja on useita ja se valitaan neuroverkon rakenteen ja käyttötarkoituksen mukaan.

Oppiminen voidaan jakaa *suunnattuun* (supervised learning) ja *suuntaamattomaan oppimiseen* (unsupervised learning). Suunnatussa oppimisessa jokaiselle syötteelle on olemassa haluttu tulos, jota kohti neuroverkkoa yritetään opettaa. Suunnatussa opetuksessa tiedetään jokaiselle syötteelle mikä sen tulos tulisi olla [Callan, 1999]. Suuntaamattomassa oppimisessä syötteillä ei ole haluttuja tuloksia, jolloin neuroverkko itse määrittää neuroverkon tulokset. Lisäksi on olemassa *vahvistava oppiminen* (reinforcement learning, graded learning), jota käytetään edellisiä oppimisia vähemmän neuroverkoissa.

Neuroverkon opettamiselle on asetettava tiettyjä lopetusehtoja, joiden mukaan neuroverkon opettaminen lopetetaan. Ilman lopetusehtoja opetus ei aina loppuisi. Kun opetus menee kerran läpi koko opetusaineiston, sanotaan sitä yhdeksi *epookiksi* (epoch) [Swingler, 1996]. Opetuksessa käytävien epookkien määrän rajoittaminen on yleisin opetuksen lopetusehto, jota voidaan käyttää sekä suunnatussa että suuntaamattomassa opetuksessa. Suunnattujen opetusten yhteydessä lopetusehtona on yleensä myös *keskineliövirheen neliöjuuri* (root mean square error) tai joku muu tapa laskea neuroverkon tekemä virhe. Keskineliövirheen neliöjuuri on neuroverkon antamien virheiden neliöiden summien neliöjuuri

$$e = \sqrt{\frac{1}{N} \sum_{i=1}^N (o_i - d_i)^2}, \quad (2.7)$$

missä o_i on neuroverkon antama tulos, d_i on haluttu tulos ja N on tulosneuronien lukumäärä [Juhola, 2000]. Kun neuroverkon keskineliövirheen neliöjuuri menee ennalta määrätyn keskineliövirheen neliöjuuren alle,

lopetetaan opettaminen. Neuroverkon sanotaan suppenevan, kun sen keskineliövirheen neliöjuuri pienenee. Muita käytettyjä lopetusehtoja ovat mm. opetukseen käytettävä aika ja keskineliövirheen neliöjuuren muutos, eli pienin gradientti (derivaatta).

On olemassa kaksi erilaista tapaa antaa tapaukset neuroverkolle opetuksessa: *tapaus* (pattern) ja *ryhmä* (batch). Ryhmätavassa yhden epookin aikana otetaan muistiin jokaisen tapauksen virheet ja päivitetään neuroverkon painoarvoja sen ylös ottamien virheiden keskiarvon perusteella. Tapaustavassa neuroverko päivittää painoarvoja jokaisen tapauksen jälkeen. Ei ole mitään sääntöä kumpaa tapaa opetuksessa tulisi käyttää, vaan se on valittava tapauskohtaisesti. Ryhmätapa on nopeampi kuin tapaustapa, koska siinä päivitetään harvemmin neuroverkon painoarvoja. Toisaalta ryhmätapa juuttuu helpommin paikallisiin minimeihin kuin tapaustapa. Jos opetetaan tapaustavalla, olisi hyvä vaihtaa jokaisella epookilla tapausten järjestystä satunnaisesti [Haykin, 1994]. [Swingler, 1996]

2.5.1. Suunnattu opettaminen

Juholan [2000] mukaan suunnattu opettaminen noudattaa seuraavia vaiheita:

1. Aseta satunnaiset arvot synapsien painoarvoille.
2. Esitä syöte.
3. Laske neuronien tulokset.
4. Muuta synapsien painoarvoja vahvistaen oikeita päätöksiä ja heikentämällä vääriä.
5. Jos lopetuskriteerit täyttyvät, niin lopeta opettaminen, muuten ota seuraava syöte ja mene kohtaan 2.

Kohta 4 kuvaa tyypillisesti suunnattua opettamista. Synapsien painoarvoja ja neuronien kynnysarvoja pystytään muuttamaan lähemmäksi oikeaa ratkaisua, sillä haluttu tulos tiedetään. Tällöin pystytään laskemaan neuroverkon antama virhe ja sen perusteella uudet painoarvot.

Suunnatun opetuksen voidaan kuvitella kulkevan $(n+1)$ -ulotteisella pinnalla, missä n on neuroverkon painoarvojen lukumäärä. $(n+1)$:s ulottuvuus pinnalla kuvaa neuroverkon virhettä kyseisillä painoarvoilla. Pintaa voidaan havainnollistaa kuvan 13 kaltaisella kolmiulotteisella pinnalla, missä korkeus kuvaa neuroverkon virhettä. Opetuksen tavoitteena on löytää tämän pinnan globaali minimi etsimällä oikeat painoarvot (vrt. oikeiden koordinaattien löytäminen). Opetusta voidaan havainnollistaa pallon liikkeellä pinnalla [Juhola, 2000]. Aluksi pallo asetetaan satunnaiseen pisteeseen (vrt. kohta 1) ja pallo liikkuu kohti paikallista minimiä. Aina pallo ei liiku globaaliin minimiin.

Esimerkiksi pallo voi mennä kuvassa 13 olevaan vasempaan kuoppaan, vaikka oikealla oleva kuoppa on globaali minimi. Opetuksia on yleensä toistettava, jotta löydetään parhaat painoarvot ja paras neuroverkko. Tämä vastaa globaalia minimiä pinnalla. Oletetaan että ongelman ratkaisuna on löytää painoarvot, joilla virhefunktio antaa pienimmän arvon, joka vastaa minimiä virhepinnalla [Rojas, 1996].



Kuva 13. Neuroverkon virhettä havainnollistava virhepinta. Pinnan oikealla oleva kuoppa kuvaa parasta mahdollista neuroverkkoa.

Käydään esimerkkinä usein yksikerroksisissa perceptroneissa käytetty opetusalgoritmi *Widrow-Hoff-delta sääntö* (Widrow-Hoff delta rule) Callanin [1999] mukaan. Aluksi synapsien painoarvoille on annettu pienet satunnaiset arvot. Lasketaan tulosneuronin j liittyvät uudet painoarvot w . Oletetaan tulosneuronin antaneen tuloksena arvon o_j ja haluttu tulos on d_j . Nyt tulosneuronin virhe, merkitään δ , voidaan laskea kaavalla

$$\delta_j = d_j - o_j. \quad (2.8)$$

δ_j kertoo kuinka paljon neuronin j antama tulos o_j poikkesi halutusta tuloksesta d_j . Olkoon tulosneuronin tuleva signaali x , jolloin kunkin painoarvon w_{ij} korjaus Δw_{ij} synapsilta i tulosneuronille j saadaan kaavasta

$$\Delta w_{ij} = \delta_j \eta x_i, \quad (2.9)$$

missä η on korjauskerroin väliltä $[0, 1]$. Korjauskerroin hidastaa verkon muutosta, jolloin verkko tekee pienempiä muutoksia kohti ratkaisua [Juhola, 2000]. Tämän jälkeen pystytään laskemaan uudet painoarvot kaavalla

$$w_{ij} = \Delta w_{ij} + w_{ij}. \quad (2.10)$$

Opetusalgoritmia käydään läpi niin kauan kunnes mikään opetuksen loputusehto ei ole voimassa.

Yksikerroksisen perceptronin oppiminen vektorinäkökulmasta voidaan tehdä samalla tavalla kuin Widrow-Hoff-delta sääntö. Tässä on kaksi luokkaa $A \in \mathbb{R}^n$ ja $B \in \mathbb{R}^n$, missä n on vektorien ulottuvuus. Alussa painoarvovektori w on satunnaisesti valittu. Jos vektori $x \in A$ antaa tuloksen $w \cdot x < 0$, on näiden vektoreiden välinen kulma suurempi kuin 90 astetta. Nyt painoarvovektoria on käännettävä vektorin x suuntaan, jotta vektori x luokiteltaisiin oikein. Tämä saadaan lisäämällä painoarvovektoriin vektori x . Vastaavasti jos vektori $x \in B$

antaa tuloksen $w \cdot x > 0$, on näiden vektoreiden välinen kulma pienempi kuin 90 astetta. Tällöin painoarvovektoria on käännettävä pois päin vektorin x suunnasta, mikä saadaan aikaiseksi vähentämällä x painoarvovektorista. Toistamalla tätä saavutetaan ratkaisu, jos luokat ovat lineaarisesti ratkeavia. [Rojas, 1996]

Edellä mainittu yksikerroksisen perceptronin opettaminen vektorinäkökulmasta voidaan muokata identtiseksi Widrow-Hoff-delta säännön kanssa. Vektorinäkökulmassakin voidaan päivittää painoarvovektoria korjauskertoimen η avulla, jolloin painoarvovektori muuttuu hitaammin. Myös virhe δ voidaan laskea samalla tavalla kuin Widrow-Hoff-delta säännössä, kun toisen luokan arvo on 0 ja toisen 1. Tällöin painoarvovektorin päivityssääntö on sama kuin Widrow-Hoff-delta säännössä:

$$w = \Delta w + w = \delta \eta x + w = (d - o) \eta x + w. \quad (2.11)$$

Jos neuroverkossa on piilokerroksia, niin on myös pystyttävä opettamaan piiloneuroneja, eikä vain tulosneuroneja. Piiloneuronien opettamisen tekee vaikeaksi, että niillä ei ole haluttuja tuloksia, kuten tulosneuroneilla. Tuloksen erotuksesta haluttuun arvoon pystytään laskemaan, kuinka paljon painoarvoja tarvitsee muuttaa. Tällaisten neuroverkkojen, kuten esimerkiksi monikerroksinen perceptron, opettamista varten käytetään usein *takaisinlevitysovetussääntöä* (back propagation). Takaisinlevitys-algoritmi tekee kaksi kierrosta käydessään läpi neuroverkon synapsien painoarvoja: ensiksi se menee neuroverkon läpi eteenpäin laskien neuroneiden tulokset ja sen jälkeen taaksepäin korjaten synapsien painoarvoja [Callan, 1999]. Nimi takaisinlevitys tulee siitä kun algoritmi tullessa taaksepäin levittää virheiden arvot ympäri neuroverkkoa.

Takaisinlevityssääntö etsii virhefunktion minimiä käyttäen *laskeutuvaa gradienttia* (descent gradient). Takaisinlevityssääntö laskee virhefunktion gradientin ja pyrkii sen perusteella löytämään virhefunktion minimin (vrt. pallon liikkuminen pinnalla). Virhefunktio on yleensä neuroverkon tulosten ja haluttujen tulosten funktio. Koska takaisinlevityssäännössä lasketaan virhefunktion gradienttia, on taattava virhefunktion jatkuvuus ja derivoituvuus. Koska neuroverkko koostaa aktivaatiofunktioiden tulokset, niin derivoituva aktivaatiofunktio tekee virhefunktioistakin derivoituvan. Tällöin aktivaatiofunktion on oltava jatkuva ja derivoituva, esimerkiksi sigmoidi funktio. Sigmoidi funktio antaa myös tiedon kuinka paljon painoarvoja on korjattava, kun taas Heavisiden funktio kertoo vain onko ratkaisu oikea vai väärä. [Rojas, 1996]

Takaisinlevittämissä säännöissäkin käytetään samoja suunnatun opettamisen vaiheita kuin Widrow-Hoff-delta säännössä. Ainoa merkittävä ero Widrow-

Hoff-delta sääntöön tulee kohdassa 4. Uudet synapsien painoarvot neuronilta i neuronille j lasketaan samalla tavalla kuin delta-säännössä kaavalla

$$w_{ij} = w_{ij} + \delta_j \eta o_j. \quad (2.12)$$

Tulosneuroneille tämä painoarvojen päivitys onnistuisi, koska niille pystytään suoraviivaisesti laskemaan niiden virhe δ . Piiloneuroneille virheen laskeminen ei onnistukaan yhtä helpolla kuin tulosneuroneille, koska niillä ei ole haluttua tulosta, kuten tulosneuroneilla. Kuitenkaan tulosneuronien virhettä ei tulisi laskea, kuten Widrow-Hoff-delta säännössä. Tällöin piiloneuroniin liittyviä painoarvoja ei pystyttäisi muokkaamaan ollenkaan ja monikerroksinen perceptron toimisi yksikerroksisen perceptronin tavoin [Bishop, 1995].

Edellä huomattiin, että Widrow-Hoff-delta säännöllä ei pystytä saavuttamaan monikerroksisen perceptronin etuja yksikerroksiseen perceptroniin verrattuna. Tätä varten täytyy osata laskea sekä piilo- että tulosneuronien virheet ja osata levittää virheet ympäri verkkoa. Johdetaan Juholan [2000] esittämällä tavalla takaisinlevityssäntö. Aluksi määritellään virhefunktio

$$E = \frac{1}{2} \sum_j (d_j - o_j)^2, \quad (2.13)$$

missä j on tulosneuronin indeksi, d on haluttu arvo ja o on neuroverkon antama tulos. Opetuksen tarkoituksena on pienentää virhettä E . Virhefunktiossa oleva $\frac{1}{2}$ tekee virhefunktion derivaatasta yksinkertaisemman ja helpottaa tulevaa laskentaa. Neuronin j *net* saadaan laskettua kaavasta

$$net_j = \sum_i w_{ij} o_i. \quad (2.14)$$

Tässä w_{ij} on ko. neuroniin neuronista i tuleva synapsi ja o_i sen antama tulos. Neuronin j antama tulos saadaan syöttämällä net_j neuronin aktivaatiofunktiolle f_j

$$o_j = f_j(net_j). \quad (2.15)$$

Kuten jo aikaisemmin mainittiin, on aktivaatiofunktion oltava jatkuva ja derivoituva. Nyt voidaan ketjusäännön perusteella kirjoittaa virhefunktion osittaisderivaatta painoarvon suhteen muotoon

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}}. \quad (2.16)$$

Nyt kaavan (2.16) jälkimmäinen osa saadaan muotoon

$$\frac{\partial net_j}{\partial w_{ij}} = \frac{\partial \sum_k w_{kj} o_k}{\partial w_{ij}} = \sum_k \frac{\partial w_{kj}}{\partial w_{ij}} o_k = \frac{\partial w_{1j} o_1}{\partial w_{ij}} + \dots + \frac{\partial w_{ij} o_i}{\partial w_{ij}} + \dots = o_i. \quad (2.17)$$

Märitellään virheen muutos verkon syötteiden muutoksen funktiona

$$-\frac{\partial E}{\partial net_j} = \delta_j. \quad (2.18)$$

Kun sovelletaan kaavaan (2.16) kaavoja (2.17) ja (2.18) saadaan virhefunktio muotoon

$$-\frac{\partial E}{\partial w_{ij}} = \delta_j o_i. \quad (2.19)$$

Virhefunktion arvon vähentäminen merkitsee painoarvojen muutosten suhteuttamista arvoon δ_j , ts. (vrt. kaava (2.12))

$$\Delta w_{ij} = \eta \delta_j o_i. \quad (2.20)$$

Nyt kun neuronin j tekemä virhe δ_j tunnetaan, niin pystytään pienentämään neuroverkon antamaa virhettä E . Muokataan neuronin j virhe δ_j ketjusäännön avulla kaavasta (2.18) muotoon

$$\delta_j = -\frac{\partial E}{\partial net_j} = -\frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial net_j}. \quad (2.21)$$

Nyt voidaan soveltaa kaavan (2.21) jälkimmäistä osaa. Koska neuronin j tulos o_j on net_j :n funktio, päästään kaavan (2.21) jälkimmäisestä osasta kaavan (2.15) perusteella muotoon

$$\frac{\partial o_j}{\partial net_j} = f_j'(net_j). \quad (2.22)$$

Kaavassa (2.22) tulee viimeistään selväksi, että aktivaatiofunktion on oltava derivoituva. Muokataan seuraavaksi kaavan (2.21) ensimmäinen osa. Kun virhefunktio E (2.13) derivoidaan neuronin j tuloksen o_j suhteen, saadaan se muotoon

$$\frac{\partial E}{\partial o_j} = -(d_j - o_j). \quad (2.23)$$

Kaavassa (2.23) voidaan huomata virhefunktion E (2.13) kertoimen $\frac{1}{2}$ etu. Nyt käyttämällä kaavoja (2.22) ja (2.23) kaavaan (2.21) saadaan

$$\delta_j = f_j'(net_j)(d_j - o_j). \quad (2.24)$$

Kaavaa (2.24) voidaan käyttää tulosneuronien virheen laskemiseen, koska niille tiedetään haluttu tulos d .

Piiloneuronien virhettä ei kuitenkaan pystytä laskemaan kaavalla (2.24), koska niillä ei ole haluttua tulosta d . On siis vielä johdettava miten virhe δ pystytään laskemaan piiloneuroneille. Kirjoitetaan ketjusäännön avulla kaava (2.23) käyttäen kaavaa (2.14) muotoon

$$\frac{\partial E}{\partial o_j} = \sum_k \frac{\partial E}{\partial net_k} \frac{\partial net_k}{\partial o_j} = \sum_k \frac{\partial E}{\partial net_k} \frac{\partial}{\partial o_j} \sum_i w_{ik} o_i. \quad (2.25)$$

Nyt kaava (2.25) saadaan muokattua muotoon

$$\sum_k \frac{\partial E}{\partial net_k} \frac{\partial}{\partial o_j} \sum_i w_{ik} o_i = - \sum_k \delta_k \left(\frac{\partial w_{1k} o_1}{\partial o_j} + \dots + \frac{\partial w_{jk} o_j}{\partial o_j} + \dots \right) = - \sum_k \delta_k w_{jk} . \quad (2.26)$$

Yhdistämällä kaavat (2.25) ja (2.26) saadaan

$$\frac{\partial E}{\partial o_j} = - \sum_k \delta_k w_{jk} . \quad (2.27)$$

Kun sijoitetaan kaavaan (2.21) kaavat (2.22) ja (2.27) saadaan

$$\delta_j = f_j'(net_j) \sum_k \delta_k w_{jk} . \quad (2.28)$$

Kaavaa (2.28) voidaan käyttää piiloneuronien virheen laskemiseen. Virhe lasketaan ko. piilokerrosta seuraavan kerroksen virheiden perusteella. Ensin on siis laskettava tulosneuronien virheet kaavalla (2.24), jonka jälkeen mennään läpi piilokerrokset taaksepäin laskien virheet kaavalla (2.28).

Ennen kuin pystyy laskemaan neuronin virheen, δ on tiedettävä neuronin aktivaatiofunktion derivaatta. Esimerkiksi logaritmisen sigmoidi funktion (2.4) derivaatta saadaan yksinkertaiseen muotoon

$$f_j'(net_j) = \frac{e^{-net_j}}{(1+e^{-net_j})^2} = f_j(net_j)(1-f_j(net_j)) = o_j(1-o_j) , \quad (2.29)$$

joka saadaan helposti neuronin tuloksesta. Vastaavasti tangenttisen sigmoidi funktion (2.5) derivaatta saadaan yksinkertaiseen muotoon

$$f_j'(net_j) = \frac{4e^{-2net_j}}{(1+e^{-2net_j})^2} = 1-f_j(net_j)^2 = 1-o_j^2 . \quad (2.30)$$

Virhettä δ laskettaessa on otettava siis huomioon aktivaatiofunktion derivaatta, koska se on erilainen jokaisella aktivaatiofunktiolla.

Käydään vielä läpi miten tulos- ja piiloneuronien virheet lasketaan, jos neuronien aktivaationfunktio on logaritminen sigmoidi (2.4). Virhetermi δ_j tulosneuronille j lasketaan tällöin kaavalla

$$\delta_j = o_j(1-o_j)(d_j - o_j) \quad (2.31)$$

ja piiloneuronille k kaavalla

$$\delta_k = o_k(1-o_k) \sum_j \delta_j w_{kj} . \quad (2.32)$$

Virheen δ perusteella pystytään päivittämään neuroverkon painoarvoja kaavalla (2.12). Mikäli neuronin aktivaatiofunktio on eri kuin logaritminen sigmoidi, niin virhetermit (kaavat (2.31) ja (2.32)) lasketaan eri tavalla riippuen aktivaatiofunktion derivaatasta.

Juholan [2000] mukaan suunnatussa opetuksessa ensimmäiseksi alustetaan neuroverkon painoarvot pienillä satunnaisluvuilla, kuten jo luvun alussa mainittiin. Jos painoarvot alustetaan liian pienillä luvuilla tai nolilla, niin tulosneuroneilta piiloneuroneille takaisin levitetty virhe tulee olemaan erittäin

pieni [Rojas, 1996]. Tämä sen takia, että laskettaessa piiloneuronien virheitä ne lasketaan käyttäen seuraavan kerroksen painokertoimilla kerrottujen virheiden avulla (vrt. kaava (2.32)). Tällöin pienet painoarvot vähentävät saatavan virheen suuruutta merkittävästi ja hidastavat neuroverkon oppimista. Etenkin jos neuroverkon kaikki painokertoimet ovat nolliä, niin tällöin kaikkien piiloneuroneiden virhe tulisi olemaan nolla. Tällöin neuroverkko pystyisi oppimaan vain lineaarisia ratkaisuja, kuten yksikerroksinen perceptron, koska vain tulosneuronien painoarvoja päivitettäisiin. Yleisesti olisi hyvä alustaa painoarvot satunnaisluvulla joltain väliltä $[-\alpha, \alpha]$ [Rojas, 1996]. Mitään parasta α :n arvoa ei kuitenkaan ole, vaan se on yleensä etsittävä tapauskohtaisesti [Rojas, 1996].

Mitä pienempi korjauskerroin η on, sitä pienempiä neuroverkon painoarvoihin tehtävät muutokset ovat ja niiden muutos on pehmeämpi. Tämä muutos hidastaa selvästi neuroverkon oppimista ja pienillä korjauskertoimilla opetus jää helpommin paikallisiin minimeihin [Haykin, 1994]. Toisaalta liian suuri korjauskerroin neuroverkon nopean oppimisen toivossa saattaa johtaa siihen, että oppimisesta tulee epävakaa eikä neuroverkko opi [Haykin, 1994]. Esimerkiksi jos jalkapallossa potkaisee pallon vähän ohi maalin, niin ei kannata korjata montaa metriä potkua maalia kohti. Tällöin riittää parin metrin korjaus. Toisaalta jos potku menee monta metriä ohi, niin parin metrin korjaus ei riitä. Eräs keino korjauskertoimen suuruuden määrittelemiseen on aloittaa suuresta arvosta ja pienentää sitä vähitellen opetuksen edetessä. Kun alussa käytetään suurta korjauskerrointa, niin otetaan suuria askeleita optimaalista ratkaisua kohti. Korjauskertoimen pienentäminen vähitellen auttaa, että opetus jää mahdollisesti minimiin, eikä lähde sieltä enää pois. Tällä tavalla voidaan ohittaa lokaaleja minimejä ja löytää mahdollisesti optimaalinen ratkaisu [Juhola, 2000].

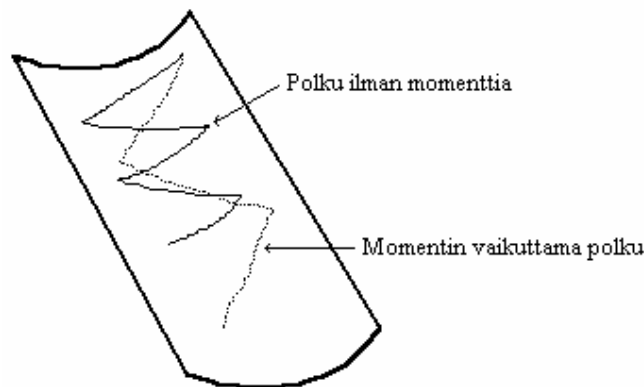
Haykinin [1994] mukaan jokaisen neuroverkon kerroksen tulisi oppia samalla asteella. Yleensä viimeisten kerrosten virheet ovat suurempia kuin ensimmäisten kerrosten virheet. Siksi viimeisten kerrosten vahvistuskertoimen tulisi olla pienempi kuin ensimmäisten kerrosten vahvistuskerroin. Lisäksi neuronit joihin tulee enemmän syötteitä, tulisi olla pienempi korjauskerroin kuin neuroneilla joilla on vähän syötteitä. [Haykin, 1994]

Neuroverkon oppimista nopeuttamaan voidaan ottaa käyttöön *momenttitermi* (momentum) α . Momentti nopeuttaa neuroverkon oppimista antaen neuroverkon painoarvojen kasvaa tai vähentyä nopeammin. Momentti antaa neuroverkon reagoida virhepinnan viimeisiin suuntauksiin [MATLAB, 2002]. Momentti auttaa neuroverkkoa oppimaan nopeammin etenkin kun opetus menee pitkää kuilua pitkin. Tällöin momentti auttaa opetusta

edistymään nopeammin kuilua pitkin, sillä virhepinnan suunta on kuilun suuntainen. Kuva 14 havainnollistaa momentin käyttöä energiapinnan pitkässä kourussa. Momenttitermin kanssa neuroverkon painokertoimia päivitetään hetkellä t kaavalla

$$w(t)_{ij} = w(t-1)_{ij} + \delta_j \eta o_j + \alpha(w(t-1)_{ij} - w(t-2)_{ij}). \quad (2.33)$$

Aikaisempaan painoarvojen päivytykseen erona on nykyisen painoarvon ja edellisen painoarvon erotuksen lisääminen kerrottuna momentilla α . Momentin ollessa 0 se on sama kuin normaali painoarvojen päivytys. Swingler [1996] suosittelee asettamaan momentin arvoksi 0,9.



Kuva 14. Momentin vaikutus energiapinnalla pitkässä kourussa. Momentin vaikuttama polku (katkoviiva) laskee selvästi nopeammin kuin polku ilman momenttia. [Juhola, 2000]

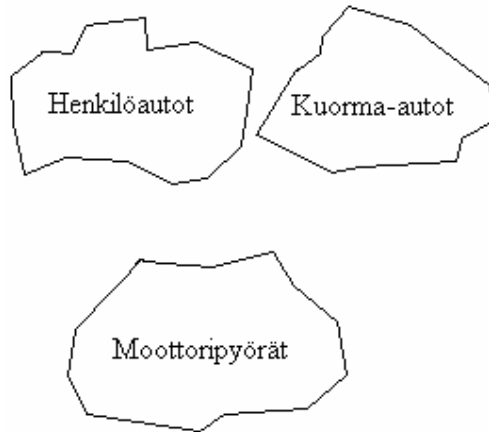
2.5.2. Suuntaamaton opettaminen ja SOM

Suuntaamattomassa oppimisessa neuroverkko asettaa saamansa syötteet muodostamiinsa *ryhmiin* (cluster). Suuntaamattoman opettamisen muodostamien ryhmien tulisi täyttää seuraavat kaksi ominaisuutta [Callan, 1999]:

1. Ryhmässä olevien syötteiden tulisi olla samantyyllisiä.
2. Ryhmät, jotka ovat samantyyllisiä, tulisi olla lähellä toisiaan.

Kuvitellaan että neuroverkko saisi syötteenään erilaisia moottoriajoneuvoja. Tällöin opetettu neuroverkko voisi tehdä esimerkiksi henkilöautoista oman ryhmän, kuorma-autoista oman ryhmän ja moottoripyöristä oman ryhmän (kuva 15). Tällöin esimerkiksi jokainen moottoripyörä ryhmiteltäisiin samaan ryhmään (ominaisuus 1). Koska henkilö- ja kuorma-auton ominaisuudet ovat lähellä toisiaan, niin niiden muodostamat ryhmät olisivat lähempänä toisiaan kuin moottoripyörien muodostama ryhmä niitä (ominaisuus 2). Yleisin

suuntaamaton neuroverkko on Teuvo Kohosen *itseorganisoituva kartta* (self-organizing map), eli *SOM-kartta*.



Kuva 15. Esimerkki kolmen eri ryhmän ryhmittelystä.

SOM-kartan algoritmissa synapsien uudet painoarvot lasketaan erilailla kuin aiemmin esitellyissä Widrow-Hoff-delta säännössä ja takaisinlevityssäännössä. SOM-kartassa on syöte- ja tuloskerros ja neuroneilla ei ole kynnyksarvoja. Jokaisesta syötoneuronista menee synapsi tulosneuroniin. SOM-kartassa voidaan ajatella kunkin tulosneuronin muodostavan oman ryhmän ja kukin syöte ryhmittyy johonkin näistä tulosneuroneista. SOM-algoritmissa käydään läpi syötteitä x ja päivitetään aina tulosneuroneista esimerkiksi *euklidisen etäisyyden neliön* (euclidian squared distance) (2.34) mukaan lähimmän neuronin (voittajaneuronin) ja sen naapuruston painoarvoja.

Käydään vielä tarkemmin läpi SOM-algoritmi. Algoritmissa lasketaan jokaisen kierroksen aluksi jokaisen tulosneuronin j euklidisen etäisyyden neliö d_j syötteeseen x , nähden

$$d_j = \sum_{i=0}^{n-1} (x_i - w_{ij})^2. \quad (2.34)$$

Tämän jälkeen lasketaan uudet painoarvot voittajaneuronin j^* (so. syötteelle x lähin tulosneuroni j) ja sen naapuruston, jonka koko hetkellä t on $N_{j^*}(t)$, painoarvoille kaavalla

$$w_{ij} = w_{ij} + \eta(x_i - w_{ij}). \quad (2.35)$$

Tässäkin η toimii vahvistuskertoimena, kuten Widrow-Hoff-delta säännössä. Naapuruston koko $N_{j^*}(t)$ pienenee ajan kuluessa rajoittaen aktivaation alaa. SOM-algoritmi tuo jokaisella askeleella neuronin i eniten vastaavan tulosneuronin j painoarvoja lähemmäs. [Juhola, 2000]

Juhola [2000] sanoo yhteenvetona SOM-algoitmista kaksi asiaa. Etsi syötettä lähin täsmäävä tulosneuron ja kasvata tämän neuronin ja sen naapurustossa olevien painoarvojen samanlaisuutta syötteeseen nähden.

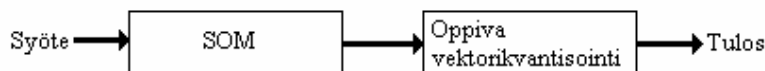
2.5.3. Muita opetustapoja

On olemassa myös muita opetustapoja kuin ohjaava ja ohjaamaton oppiminen. Tavallisesti nämä ovat näiden pieniä muunnoksia tai yhdistelmiä.

Vahvistava oppiminen on samanlainen kuin ohjattu oppiminen paitsi siinä neuroverkko saa vain korjauksen suunnan tai tuloksen tarkkuuden eikä tarkkaa haluttua tulosta [Hech-Nielsen, 1990]. Opetuksessa voidaan esimerkiksi kertoa vain oliko tulos oikea vai väärä, eikä kerrota kuinka paljon tulos oli väärä.

Takaisinlevityssääntöön verrattuna vahvistavan oppimisen ei tarvitse laskea monimutkaisia gradientteja, mikä tekee siitä käyttökelpoisen monimutkaisissa järjestelmissä. Toisaalta vahvistava oppiminen on puutteellinen suuriin järjestelmiin. Lisäksi vahvistava oppiminen voi juuttua helposti paikallisiin minimeihin. [Fu, 1994]

On myös olemassa opetustapoja, joissa yhdistetään ohjattua ja ohjaamatonta oppimista. Eräs tällainen on *oppiva vektorikvantisointi* (learning vector quantisation). Oppivassa vektorikvantisoinnissa muodostetaan aluksi SOM-kartta. Tämän jälkeen SOM-karttaa muokataan ohjatulla oppimisella. Oppivan vektorikvantisoinnin prosessia kuvaa kuva 16. [Haykin, 1994]



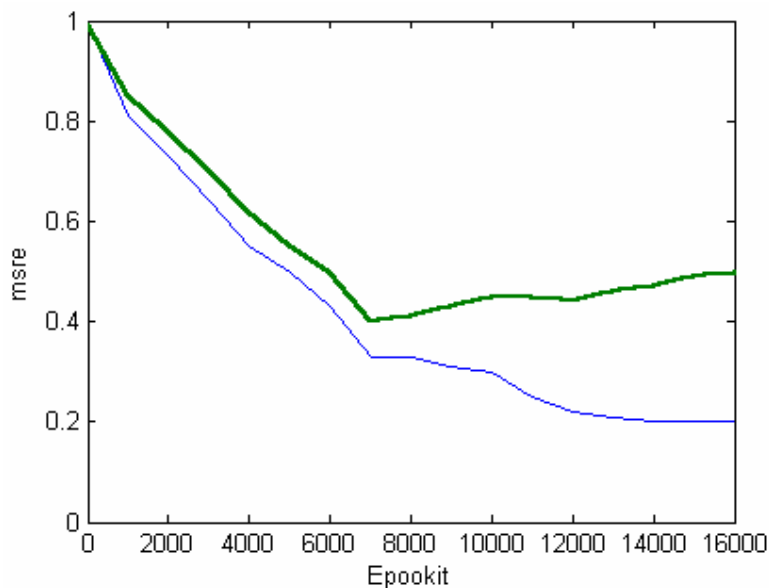
Kuva 16. Oppivan vektorikvantisoinnin prosessi. [Haykin, 1994]

2.5.4. Neuroverkon ylioppiminen ja arkkitehtuuri

Neuroverkkojen tarkoituksena on *yleistää* (regularisation) oppimaansa aineistoa. Yleistämisellä tarkoitetaan neuroverkon kykyä antaa oikeita tuloksia aineistolle, jolla sitä ei ole opetettu [Callan, 1999]. Suunnatussa oppimisessa neuroverkolla on vaarana *ylioppia* (over-fit) opetusaineisto, jolloin se ei osaa yleistää. Ylioppinut neuroverkko osaa hyvin opetusaineiston, mutta muuta aineistoa se ei osaa käsitellä oikein. Ylioppimalla heikennetään neuroverkon kykyä yleistää [Picton, 2000]. Neuroverkon ylioppimiseen voi vaikuttaa kaksi asiaa: epookkien määrä ja piiloneuroneiden määrä. Kuvat 18 ja 19 havainnollistavat neuroverkkojen ylioppimista ja yleistämistä.

Neuroverkon opetuksen keskineliövirheen neliöjuuri kertoo miten hyvin neuroverkko on oppinut opetusaineistoa. Opetuksen tavoitteena on vähentää

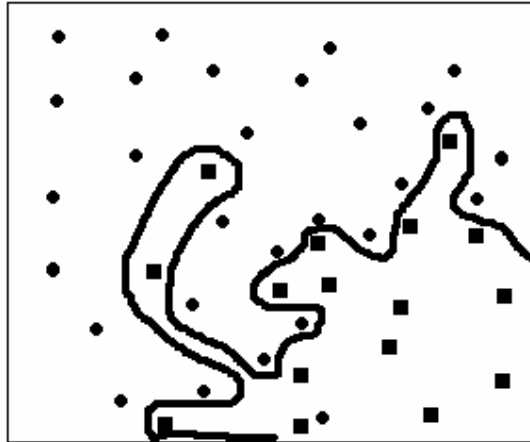
keskineliövirheen neliöjuurta. Keskineliövirheen neliöjuuren liiallinen pienentäminen aiheuttaa usein neuroverkon ylioppimisen, sillä neuroverkko oppii tällöin liian tarkasti opetusaineiston. Jos neuroverkkoa opettaa liikaa, eli liian monta epookkia, niin neuroverkko ylioppii helposti. Sopivan epookkien määrän valitseminen on vaikeaa. Eräs tapa löytää neuroverkon mahdollinen ylioppimispiste on käyttää erikseen validointiaineistoa, joka on erillinen opetus- ja testiaineistosta. Validointiaineisto annetaan opetuksessa jokaisen epookin jälkeen neuroverkolle. Tiettyyn epookkiin asti validointiaineiston antama keskineliövirheen neliöjuuri laskee, mutta lakkaa tämän jälkeen laskemasta tai alkaa nousta. Tämän epookin jälkeen neuroverkko alkaa ylioppia ja se on saavuttanut parhaan tuloksensa. Kuvasta 17 näkee miten validointiaineisto toimii neuroverkon opetuksessa.



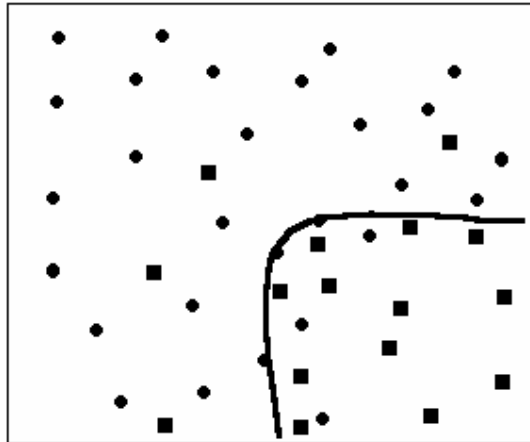
Kuva 17. Opetus- ja validointiaineiston keskineliövirheen neliöjuuren muutos opetuksen edetessä. Validointiaineisto (paksu viiva) saavuttaa miniminsä 7000 epookin kohdalla, jonka jälkeen neuroverkko alkaa ylioppia.

Neuroverkon *arkkitehtuurilla* (architecture) on myös iso merkitys mahdolliseen ylioppimiseen. Neuroverkon arkkitehtuurilla tarkoitetaan neuroverkossa käytettävien neuroneiden ja kerrosten määriä. Yleensä syöte- ja tulosneuronien määrät määräytyvät ongelman mukaan, mutta piiloneuronien määrälle ei ole selkeää sääntöä. Suuremmilla piiloneuronien ja -kerrosten määrillä neuroverkko pystyy mallintamaan monimutkaisempia ongelmia, kuten jo aikaisemmin on todettu. Piiloneuronien määrän ei tulisi olla liian suuri. Liian monen piiloneuronin vaarana on, että neuroverkko ylioppii, kuten esimerkiksi kuvassa 18 on käynyt. Sopivalla arkkitehtuurilla opetettu neuroverkko osaisi yleistää kuvan 19 tapaan. Mitä enemmän piiloneuroneita

neuroverkossa on, sitä suuremmalla riskillä neuroverkko ylioppii. Callan [2000] antaa ylioppimisesta esimerkkinä neuroverkon, jossa on yhtä monta piiloneuronia kuin opetusaineistossa on tapauksia. Tällöin jokaista tapausta kohden olisi oma piiloneuroni, joka oppisi tunnistamaan oman tapauksen. Neuroverkko ei kuitenkaan osaisi toimia oikein saadessaan syötteen, jolla sitä ei ole opetettu, sillä piiloneuronit osaisivat toimia vain opetetuilla tapauksilla. Neuroverkko olisi ylioppinut opetusaineiston.



Kuva 18. Liian monen piiloneuronin arkkitehtuurin seurauksena on ylioppiminen, kuten tässä kuvassa on käynyt. Tulos voi olla sama, jos neuroverkkoa opetetaan liian kauan.



Kuva 19. Ratkaisu, jossa neuroverkolla on sopiva määrä piiloneuroneja ja yleistäminen onnistuu.

Neuroverkon arkkitehtuurin valitsemiselle on olemassa muutamia approksimaatioita. Nyrkkisääntönä on, että neuroverkossa voi olla korkeintaan kymmenesosa painoarvoja kuin opetusaineistossa on tapauksia voidakseen oppia [Juhola, 2000]. Swingler [1996] mainitsee, että tarvittavan opetusaineiston koko voitaisiin laskea seuraavalla kaavalla

$$\text{Opetusaineiston koko} = \frac{\text{painoarvojen määrä}}{\text{tavoitettu virhe}}. \quad (2.36)$$

Kun asettaa virheeksi 0,1, niin saadaan em. nyrkkisääntö. Opetusaineiston on oltava tarpeeksi suuri, jotta neuroverkko pystyisi oppimaan sen. Yleensä tiedetään opetusaineiston tapausten ja syöte- ja tulosneuroneiden lukumäärät, joista voidaan laskea yhden piilokerroksen arkkitehtuurin piiloneuroneiden lukumäärän suositeltu yläraja

$$h = \frac{n}{10i + 10o}, \quad (2.37)$$

missä h on piiloneuronien lukumäärä, i on syötoneuronien lukumäärä, o on tulosneuronien lukumäärä ja n on opetusaineiston tapausten lukumäärä. Esimerkiksi neuroverkossa, jossa on kolme syötoneuronia, kaksi tulosneuronia ja opetusaineistossa 800 tapausta voi nyrkkisäännön mukaan olla korkeintaan 16 piiloneuronia. Swingler [1996] mainitsee muutamia sääntöjä, joiden mukaan voidaan etsiä sopiva arkkitehtuuri erilaisille ongelmille:

- Älä käytä kahta kertaa enempää piiloneuroneja kuin on syötoneuroneja.
- Piirteiden poistamiseen tarvitaan vähemmän piiloneuroneja kuin syötteitä.
- Luokitteluun tarvittavien piiloneuroneiden lukumäärä on suoraan verrannollinen luokkien määrään.

Swingler [1996] ehdottaa parhaan arkkitehtuurin etsimiseen lähteä pienimmästä piiloneuroneiden määrästä ja suurentaa sitä kunnes neuroverkon antama keskineliövirheen neliöjuuri alkaa kasvaa. Callan [2000] sanoo nyrkkisäännöksi kokeilla piiloneuroneiden lukumääräksi 30-50 prosenttia syöteneuroneiden lukumäärästä.

2.6. Muita neuroverkkoja

Aiemmin on esitetty kaksi yleisintä neuroverkkoa, eli perceptron ja SOM. Tässä luvussa esitellään muita neuroverkkoja. Erilaisia neuroverkkoja on monia, mutta tässä esitellään muutama mielenkiintoinen neuroverkko niistä.

2.6.1. Radiaalikantafunktioneuroverkko

Radiaalikantafunktioneuroverkossa (radial bases neural network) on yksi syöte-, piilo- ja tuloskerros. Neuroverkon jokainen kerros on täysin yhdistetty. Jokaisella piiloneuronilla on funktio ϕ , joka saa syötteinään syötteiden euklidisen etäisyyden kyseiseen piiloneuroniin liittyviin synapseihin nähden. Tulosneuronit antavat tuloksena niiden *net*-arvon. Radiaalikantafunktioneuroverkossa neuroneilla ei yleensä ole kynnysarvoja.

Radiaalikantafunktioneuroverkon syöte- ja piilokerroksen välisten synapsien painot opetetaan yleensä suuntaamattomalla opetusmenetelmällä ja piilo- ja tuloskerroksen välisten synapsien painoarvot suunnatulla opetusmenetelmällä. [Callan, 1999]

2.6.2. Assosiativimuistit

Assosiativimuistit (associative memory) muistuttavat ihmisen muistia, jossa tietyt asiat saavat aikaiseksi miellelyhtymiä eli assosiaatioita. Yksi assosiativimuistin käyttökohde on kohinan poistaminen kuvasta tai tiedon tallentaminen kuvaan. Assosiativimuistissa opettaminen onnistuu yksinkertaisilla matriisien yhteen- ja kertolaskulla, joten opettaminen on helpompaa ja ei vie niin paljon aikaa kuin esimerkiksi perceptronissa. Assosiativimuistit saavat syötteekseen joko binäärisiä {0, 1} tai bipolaarisia {-1, +1} arvoja. Assosiativimuistien opettaminen on yleensä syötteiden ja niihin liittyvien tuloksien tallentamista neuroverkkoon.

Autoassosiativimuistissa (auto-associative memory) neuroverkkoon tallennetaan esimerkiksi erilaisia kuvia. Tallentamisen (opettamisen) jälkeen neuroverkolle voidaan antaa kohinaisia kuvia, jolloin neuroverkko antaa tuloksena alkuperäisen kuvan. Kuvassa 20 on esimerkki autoassosiativimuistin käyttämisestä kohinaiseen a-kirjaimeen.

Heteroassosiativimuistissa (hetero-associative memory) neuroverkolle tallennetaan syöte-tulos pareja. Tällöin jokaiseen syötteeseen liittyy jokin muu

tulos kuin itse syöte (vrt. autoassosiatiivimuisti). Heteroassosiatiivimuistia voidaan verrata ihmisen muistiin, missä esimerkiksi ihmisten nimet yhdistetään heidän kasvoihinsa [Juhola, 2000]. Kuvassa 20 on esimerkki heteroassosiatiivimuistin käyttämisestä kohinaiseen isoon a-kirjaimen saadakseen pienen a-kirjaimen.

$$\begin{array}{c} \mathbb{A} \rightarrow \mathbb{A} \\ \mathbb{A} \rightarrow \mathbf{a} \end{array}$$

Kuva 20. Ylempänä esimerkki autoassosiatiivimuistin käytöstä kohinaiseen kuvaan ja alempana sama heteroassosiatiivimuistista.

2.6.3. Hopfieldin-verkko ja Boltzmannin kone

Hopfieldin-verkko on autoassosiatiivimuistin tavoin toimiva neuroverkko. Hopfieldin-verkko toimii muistin tavoin. Siihen voidaan tallentaa esimerkiksi erilaisia kuvia. Tallentamisen jälkeen Hopfieldin-verkosta voidaan palauttaa siihen tallennettuja kuvia antamalla sille kohinaisia kuvia, kuten kuvassa 21. Verkossa kaikki neuronit on yhdistetty kaikkiin muihin neuroneihin, eli se on täysin yhdistetty. Hopfieldin-verkossa ei ole erikseen syöte- ja tulosneuroneja, vaan jokainen neuroni on sekä syöte- että tulosneuroni. Tämä tekee Hopfieldin-verkosta erilaisen verrattuna aiempiin neuroverkkoihin. Hopfieldin-verkkoja pystytään käyttämään esimerkiksi kohinan poistamiseen kuvista. [Juhola, 2000]

Kun Hopfieldin-verkko saa syötteen, niin se alkaa laskea uusia tilojaan kunnes se pääsee stabiiliin ratkaisuun. Hopfieldin-verkon antama tulos on lopuksi verkon neuroneissa. [Juhola, 2000]

$$\mathbb{A} \mathbb{A} \mathbb{A} \mathbb{A} \mathbb{A}$$

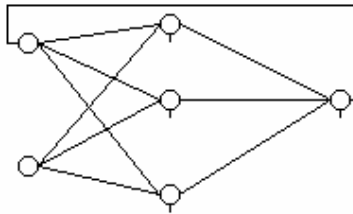
Kuva 21. Esimerkki Hopfieldin-verkon toiminnasta kohinaisen a-kirjaimen kanssa. Neuroverkko korjaa kohinaista a-kirjainta vähitellen verkolle tallennettuun muotoon.

Hopfieldin-verkossa siihen tallennetut hahmot ovat verkon paikallisissa minimeissä. Joskus Hopfieldin-verkon ratkaisut saattavat jäädä väärään paikalliseen minimiin, joka ei ole globaali minimi. Tällöin neuroverkko on juuttunut väärään ratkaisuun ja tulos ei ole oikea hahmo. Tällaista tilannetta kuvaa hyvin kuvan 13 pinta. *Boltzmannin kone* (Boltzmann machine) on Hopfieldin-verkko, jossa kuvitellulla pinnalla liikkuvaa palloa ravistellaan. Ravistuksen tarkoituksena on ehkäistä ratkaisun juuttumista paikallisiin minimeihin. [Juhola, 2000]

Verenkierrosta otetuista röntgenkuvista johdettuja kuvia käytetään sepelvaltimotautien diagnooseihin ja hoitoihin. Hopfieldin-verkkoja on käytetty verisuonien löytämiseen näistä kuvista. Siinä on annettu verkolle syöteinä harmaasävykuvista tehtyjä binäärisiä mustavalkokuvia. Opetettu Hopfieldin-verkko on poistanut kuvista vähitellen kohinan (vrt. kuva 21) ja erottanut kuvista verisuonien paikat. [Karapataki and De Wilde, 1997]

2.6.4. Takaisinkytkettyvät neuroverkot

Takaisinkytkettyvät neuroverkot (recurrent neural network, feedback neural network) ovat erikoistuneet käsittelemään sekvenssejä, joiden pituutta ei ennalta tiedetä. Takaisinkytkettyvä neuroverkko toimii muuten samalla periaatteella kuin perceptron, mutta siinä joillakin neuroneilla on synapsit takaisin itseensä tai edellisen kerroksen neuroneihin (kuva 22). [Callan, 1999]



Kuva 22. Takaisinkytkettyvä neuroverkko, jossa tulosneuronilla on takaisinkytkettyvä synapsi ensimmäiseen syöteneuroniiin.

Takaisinkytkettyvä neuroverkko pystytään esimerkiksi opettamaan lukemaan sekvenssiä kirjaimista siten, että se pystyy arvaamaan seuraavan kirjaimen kahden edellisen kirjaimen perusteella. Tällöin neuroverkolla voisi olla kaksi syöteneuronia, piilokerros, josta olisi takaisinkytkentä syötekerrokseen, ja yksi tulosneuroni.

3. Aineistot

Tässä luvussa kuvaillaan kaikki kahdeksan tutkielmassa käytettyä aineistoa, joilla neuroverkkoja on opetettu. Jokaisella aineistolla tullaan opettamaan neuroverkko. Tavoitteena on saada tehtyä jokaiselle aineistolle hyvä neuroverkko, joka osaa luokitella mahdollisimman hyvin kyseisen ongelman. Suurin osa aineistoista on kerätty UCI:n [Blake and Merz, 1998] tietokannoista. Aineistoista alokas-, huimaus- ja virtsainkontinenssi-aineisto ovat peräisin Tampereen yliopiston tietojenkäsittelytieteiden laitokselta. Näitä kolmea aineistoa on käytetty mm. kyseisen laitoksen tutkimuksissa.

Aineistojen tapausten lukumäärä vaihtelee keskimäärin noin 200 ja 800 tapauksen välillä ja attribuuttien määrä kolmen ja kolmentoista välillä. Alokas-aineisto on muihin aineistoihin verrattuna poikkeuksellisen suuri 10000 tapauksella ja 39 attribuutilla. Lisäksi jokaisessa aineistossa on yksi attribuutti luokkamuuttujana. Kunkin aineiston tiedot näkyvät taulukossa 1. Jokainen aineisto on täydellistä, eli niissä ei esiinny puuttuvia arvoja, joten ne sopivat hyvin neuroverkoille. Osaan aineistoista puuttuvat arvot on estimoitu, minkä takia nekin ovat täydellisiä. Attribuutit ovat sekä kvantitatiivisia että kvalitatiivisia ja luokkamuuttujat kvalitatiivisia. Koska kaikki luokkamuuttujat ovat kvalitatiivisia, niin jokaisen aineiston kohdalla kyseessä on luokittelu.

Taulukko 1. Aineistojen tapausten ja attribuuttien lukumäärät.

Aineisto	Tapaukset	Attribuutit
Alokas	10000	39
Buba	345	6
Ecoli	336	7
Haberman	306	3
Huimaus	815	13
New-thyroid	215	5
Pima	768	8
Virtsainkontinenssi	529	7

Neuroverkot tarvitsevat oppiakseen paljon tapauksia. Tässä tutkielmassa käytettävät aineistot ovat suhteellisen pieniä neuroverkkoja varten, jotta niistä voitaisiin luoda erikseen *opetus-* (training set) ja *testiaineistot* (test set). Opetusaineistolla opetetaan neuroverkko ja testiaineistolla testataan opetettua neuroverkkoa. Tämä on siksi, että jos neuroverkkoa testattaisiin opetusaineistolla, saataisiin liian optimistisia tuloksia. Testiaineiston avulla saadaan luotettavampi kuva opetetusta neuroverkosta ja sen yleistämiskyvystä. Tavallisesti testiaineistoon otetaan kolmas- tai neljäsosa koko aineistosta.

Koska tässä tutkielmassa käytetyt aineistot ovat liian pieniä neuroverkoille, niin opettamiseen ja testaamiseen käytetään *ristiinvalidointia* (cross-validation). Alokas-aineisto on ainoa aineisto, jolle ei tehdä ristiinvalidointia, koska se on riittävän suuri neuroverkkoja varten. Ristiinvalidoinnissa aineisto jaetaan k :hon yhtä suureen alijoukkoon. Ensimmäiset $k-1$ alijoukkoa muodostavat opetusaineiston ja k :s alijoukko testiaineiston. Jokainen alijoukko tulee toimimaan kerran testiaineistona. Tulos on kaikkien testikertojen keskiarvo. Neuroverkkojen tapauksessa tällä menetelmällä ei pystytä luomaan tiettyä neuroverkkoa, vaan sillä saadaan tietoon se, kuinka hyvin kyseisestä aineistosta pystyttäisiin tekemään neuroverkko. [Viikki, 2003]

3.1. Aineistojen esittely

Alokas-aineistoon on otettu 10000 tapauksen otos suuremmasta noin 450000 tapauksen aineistosta. Alokas-aineisto on koottu Ruotsin armeijan tekemistä tutkimuksista, joissa on tutkittu kutsunnassa olleiden kuuloa. Alokas-aineiston luokkamuuttuja kertoo onko kyseisellä henkilöllä kuulovaurioita vai ei. Alokas-aineiston opettamiseen ei ole käytetty ristiinvalidointia.

Buba-aineisto on lääketieteellinen aineisto maksan häiriöistä. Buba-aineiston jokainen tapaus koostuu mies-potilaasta saaduista tiedoista. Aineisto on peräisin vuodelta 1990 ja siinä olevien attribuuttien on uskottu olevan alttiita maksahäiriöille. Attribuuteista *mcv*, *alkphos*, *sgpt*, *sgot* ja *gammagt* ovat veritesteistä saatuja tuloksia. Buba-aineiston luokkamuuttuja *selector* kertoo onko kyseinen potilas *terve* vai *sairas*.

Ecoli-aineisto on lääketieteellinen aineisto, jossa on tarkasteltu proteiinin muodostumispaikkoja. Aikaisemmin tehdyissä tutkimuksissa ecoli-aineisto on saatu luokiteltua 81 prosentin todennäköisyydellä käyttäen *binäärisiä päätöspuita* (binary decision tree) ja *Bayesin luokittelua* (Bayesian classification) [Blake and Merz, 1998].

Haberman-aineistoon on kerätty vuosina 1958-1970 Chicagon yliopiston Billings-sairaalassa tehtyjen rintasyöpäleikkausten tietoa. Luokkamuuttuja kertoo, kuoliko potilas viiden vuoden sisällä leikkauksesta (luokka *kuoli*) vai ei

(luokka *selviytyi*). Haberman-aineisto on tässä tutkielmassa käytetyistä aineistoista yksinkertaisin.

Huimaus-aineisto käsittelee erilaisia huimaussairauksia. Aineisto on kerätty Helsingin yliopistollisessa keskussairaalaissa. Alkuperäisessä aineistossa on ollut peräti noin 150 attribuuttia, mutta tässä tutkielmassa käytettyyn aineistoon niistä on valittu 13 merkitsevintä attribuuttia. Luokkamuuttuja koostuu erilaisista huimaussairauksista, joita on kuusi tässä tutkielmassa käytetyssä aineistossa. Alkuperäisessä aineistossa on ollut yhdeksän eri huimaussairautta ja tapauksia, joiden diagnoosi on puuttunut. Tässä tutkielmassa käytettyyn aineistoon on otettu kuusi suurinta huimaussairautta alkuperäisestä aineistosta, sillä muita tapauksia oli liian vähän, jotta neuroverkko olisi pystynyt oppimaan ne. Tässä tutkielmassa käytetyssä aineistossa olevat huimaussairaudet ovat kuulohermokasvain (luokka *an*), hyvälaatuinen asentohuimaus (luokka *bpv*), Menieren tauti (luokka *Meniere*), äkillinen kuulonmenetys (luokka *sd*), traumaattinen huimaus (luokka *tv*) ja vertibulaari neuriitti (luokka *vn*). Viikki [2002] mainitsee väitöskirjassaan aikaisemmista kokeiluista luokitella huimaus-aineistoa neuroverkoilla. Tällöin luokat *bpv*, *Meniere* ja *vn* saatiin luokiteltua 90 prosentin tarkkuudella ja luokka *vs* 70 prosentin tarkkuudella. Huimaus-aineistoa on aikaisemmin yritetty luokitella *päätöspuumenetelmillä* (decision tree) päästen 89-99 prosentin tarkkuuteen [Viikki, 2002; Viikki *et al.*, 1999, 2000, 2001]. Huimaussairauksien diagnoosin helpottamiseksi on kehitetty asiantuntijajärjestelmä ONE [Auramo, 1999].

New-thyroid -aineisto käsittelee kilpirauhasen toimintaa. Luokkamuuttuja jakaa kilpirauhasen toiminnan normaaliin (*normal*), suureen (*hyper*) ja erittäin suureen (*hypo*). New-thyroid -aineisto on aikaisemmin saatu mallinnettua 100 prosentin todennäköisyydellä käyttämällä *kernel-funktioita* (kernel density method) [Blake and Merz, 1998]. New-thyroid -aineisto on pienin tässä tutkielmassa käytetyistä aineistoista.

Pima-aineisto koostuu Pima-intiaaniheimon 21 vuotta täyttäneiden naisten tiedoista [Blake and Merz, 1998]. Aineiston luokkamuuttuja kertoo, onko ko. intiaanilla diabetes. Pima-aineisto on viidestä UCI:n [Blake and Merz, 1998] aineistosta selvästi suurin. Attribuutteina on mm. raskauksien lukumäärä, painoindeksi ja ikä. Käyttäen 576 tapausta opetusaineistossa ja 192 tapausta testiaineistossa Pima-aineisto on saatu mallinnettua 76 prosentin todennäköisyydellä *ADAP-algoritmillä* [Blake and Merz, 1998]. Pima-aineisto on saatu luokiteltua *CIDIM-päätöspuualgoritmillä* 75 prosentin tarkkuudella [Jerez-Aragonés *et al.*, 2003].

Virtsainkontinenssi-aineistossa on tahattomista virtsankarkailuista kärsivien naisten potilastietoja. Aineisto on peräisin Kuopion yliopistollisesta sairaalasta vuosilta 1989-1998. Aineistossa olevien naisten ikä on ollut 26-89 vuotta ja iän keskiarvo on ollut 52 vuotta [Laurikkala *et al.*, 2001b]. Tässä tutkielmassa käytetty aineisto on vähän alkuperäistä aineistoa pienempi käsittäen 529 tapausta, seitsemän attribuuttia ja luokka-attribuutin. Alkuperäisessä aineistossa oli kuusi eri luokkaa, mutta tässä tutkielmassa käytetyssä aineistossa neljä pienintä luokkaa on yhdistetty kolmanneksi *muut* luokaksi. Luokan *muut* alkuperäisiä luokkia ei tässä tutkielmassa käytetyssä virtsainkontinenssi-aineistossa ole, vaan pienet luokat on valmiiksi yhdistetty luokaksi *muut*. Laurikkala *et al.* [2001b] ovat käyttäneet mm. *k:n lähimmän naapurin menetelmää* (*k*-nearest neighbour) aineiston luokitteluun saaden tapaukset luokiteltua n. 90 prosentin tarkkuudella.

3.2. Aineistojen muokkaaminen neuroverkkoja varten

Ennen kuin neuroverkolle syötetään aineistoa opettamista varten, on aineistoa yleensä esikäsiteltävä. Esikäsittelemätön aineisto saattaa johtaa siihen, ettei neuroverkko opi tai se oppii väärin. On olemassa useita menetelmiä joilla aineisto saadaan neuroverkolle suotuisaksi.

Neuroverkoille syötettävän aineiston on oltava täydellistä, eli siinä ei saa olla puuttuvia arvoja. Jos puuttuvia arvoja on noin alle 30 prosenttia attribuutin arvoista, voidaan erilaisilla menetelmillä estimoida puuttuvat arvot [Juhola, 2000]. Kaikki tässä tutkielmassa käytetty aineisto on täydellistä, koska puuttuvat arvot on estimoitu valmiiksi, joten puuttuvien arvojen suhteen aineistoille ei tarvitse tehdä mitään.

Aineiston tulee olla skaalattu samoille arvoväleille. Yleensä neuroverkon saamat syötteet skaalataan joko binääriseen asteikkoon [0, 1] tai bipolaariseen asteikkoon [-1, 1]. Jos arvoja ei skaalata, niin suuret attribuutit saattavat dominoida ja neuroverkko ei opi oikein. Esimerkiksi ihmisen ikä on tällöin skaalattava. Yksinkertaisin tapa skaalata attribuutin arvo x välille [0, 1] on lineaarinen skaalaus

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \quad (3.1)$$

Se skaalaa arvot lineaarisesti siten, että suurin arvo on 1 ja pienin 0 attribuutin suurimman ja pienimmän arvon perusteella. Skaalatun arvon saa palautettua takaisin alkuperäiseksi ratkaisemalla x :n samasta kaavasta. Jos arvot halutaan välille [-1, 1], niin tulos kerrotaan kahdella ja vähennetään yksi. Usein on tarpeen skaalata arvot tietylle välille. Esimerkiksi käytettäessä sigmoidi funktiota skaalaaminen välille [0.1, 0.9] vähentää rasiitetta nostaa arvoja

sigmoidi funktion ääriarvoihin, sillä sigmoidi funktio ei koskaan saavuta arvoja 0 ja 1, vaan lähestyy niitä [Callan, 1999]. Kaavalla

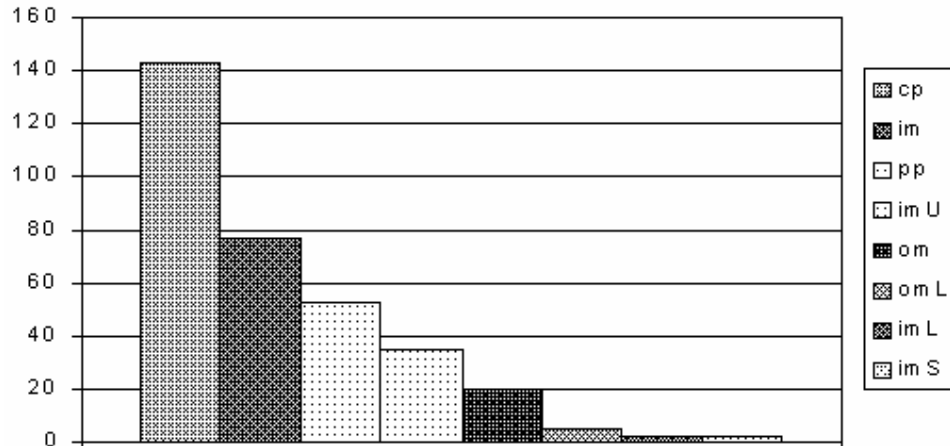
$$y = \frac{s_{\max} - s_{\min}}{x_{\max} - x_{\min}} (x - x_{\max}) + s_{\max} \quad (3.2)$$

arvot voidaan skaalata välille $[s_{\min}, s_{\max}]$ [Swingler, 1996]. Kaava (3.1) on erikoistapaus kaavasta (3.2) välille $[0, 1]$. Molemmat näistä kaavoista saattavat johtaa tapausten tiivistymiseen kasaan, mikäli joitakin arvoja on muita selvästi enemmän. Parempi tapa skaalata arvoja on standardointi

$$y = \frac{x - \bar{x}}{\sigma_x}. \quad (3.3)$$

Tässä attribuutin arvosta vähennetään ko. attribuutin keskiarvo ja jaetaan se ko. attribuutin keskihajonnalla. Standardointi skaalaa arvot tasaisin välein siten, että uusi keskiarvo on 0 [Swingler, 1996]. Lineaarisen normalisoinnin jälkeen arvot eivät välttämättä ole vielä millään välillä, vaan on vielä käytettävä esimerkiksi lineaarista skaalaamista. Jos jakauman väli on tunnettu, voidaan tietää normalisoinnin jälkeisten arvojen väli.

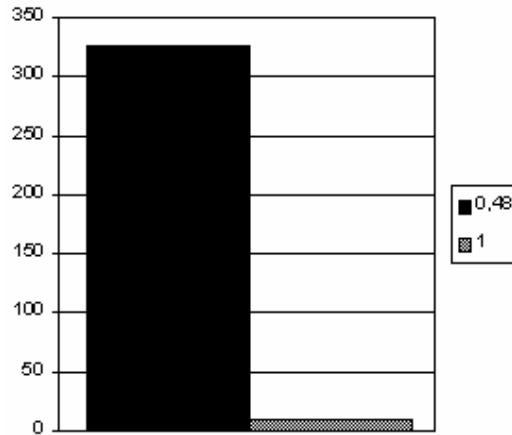
Opetusaineiston luokkamuuttujan kaikkien luokkien frekvenssi tulisi olla suurempi kuin kymmenen. Jos yhtä luokkaa on alle kymmenen, ei neuroverkko pysty oppimaan tätä luokkaa. Kun tapauksia on liian vähän, ei neuroverkko pysty oppimaan niitä. Tällöin niitä ei myöskään pystytä testaamaan luotettavasti. Kymmenkertaisen ristiinvalidoinnin tapauksessa frekvenssin on oltava vähintään 12, jotta opetusaineistossa frekvenssi olisi keskimäärin vähintään kymmenen. Vaikka testit osoittaisivatkin pienempien luokkien luokittuvan oikein, saattaa kyseessä oleva vastaus olla sattumalta tullut. Jos testaukseen jää vain pari tapausta, ei testien tulosta voida pitää luotettavana. Toinen vaihtoehto on yhdistää pienemmät luokat yhdeksi uudeksi luokaksi. Uuden luokan on tällöin täytettävä myös em. ehdot. Ecoli-aineistossa luokkien frekvenssit ovat 143 (*cp*), 77 (*im*), 52 (*pp*), 35 (*imU*), 20 (*om*), 5 (*omL*), 2 (*imL*) ja 2 (*imS*) (kuva 23). Tässä tapauksessa luokkien *omL*, *imL* ja *imS* frekvenssit ovat liian pieniä, jotta neuroverkko pystyisi oppimaan ne. Näiden luokkien tapaukset tullaan poistamaan kokonaan tässä tutkielmassa käytettävästä ecoli-aineistosta. Toisena vaihtoehtona on yhdistää luokat *om* ja *omL* tai jotkin muut luokat samantapaisina luokkina.



Kuva 23. Ecoli-aineiston luokkamuuttujan frekvenssit.

Nominaalisten muuttujien suhteen täytyy neuroverkkojen kanssa olla tarkkana. Koska neuroverkko tulkitsee syötteensä reaalityyppinä, niin on mietittävä, miten voisi ilmaista esimerkiksi siniset ja ruskeat silmät eri tavalla. Yksinkertaisin ja varmin tapa tähän on koodata muuttujat bittijonoksi. Tällöin jokainen bitti olisi oma syöte tai tulos ja esimerkiksi 1 merkisi kyseisen ominaisuuden olevan päällä ja 0 pois päältä. Bittijonon huono puoli on, että se tekee neuroverkon rakenteen monimutkaisemmaksi. Tässä tutkielmassa käytetyissä aineistoissa suurin osa syötteistä on kvantitatiivisia ja tulokset nominaalisia. Osa tuloksista on vain kaksiluokkaisia, jolloin ne voidaan esittää yhtenä bittinä. Mutta, jos luokkia on enemmän kuin kaksi, niin yhden bitin esitys ei ole mahdollinen. Esimerkiksi ecoli-aineistossa on mietittävä, miten koodata luokkamuuttuja, koska siinä on enemmän kuin kaksi luokkaa.

Neuroverkon syötteistä vakioarvoiset attribuutit tulisi poistaa. Mikäli esimerkiksi attribuutin arvo 95 prosentissa tapauksista on sama, voidaan se usein poistaa merkityksettömänä [Juhola, 2000]. Tällainen attribuutti ei lisää informaatiota ja voidaan sen takia poistaa. Turhien attribuuttien poistaminen vähentää neuroverkon syöteneuronien määrää ja täten yksinkertaistaa neuroverkon oppimista ja käyttöä. Ecoli-aineiston *lip*-attribuutin frekvenssin (kuva 24) huomataan olevan lähes vakio samoin kuin attribuutti *chg*. Nämä voidaan poistaa aineistosta ennen, kuin sillä opetetaan neuroverkkoa.



Kuva 24. Ecoli-aineiston *lip*-attribuutin frekvenssi. 97 prosenttia arvoista on 0,48, joten attribuutti voidaan poistaa aineistosta.

Kerätyssä aineistossa esiintyy usein virheellisiä arvoja. Virheellisiä arvoja voi olla aineistossa esimerkiksi mittausvirheiden tai inhimillisten erehdysten kautta. Esimerkiksi jos ihmisen paino on 600 kg, voidaan tätä pitää virheellisenä arvona. Virheelliset arvot voidaan havaita aineistosta käymällä se läpi tai laskemalla esimerkiksi 95 tai 99 prosentin luottamusväli [Viikki, 2003]. Mikäli arvo menee valitun luottamusvälin ulkopuolelle, voidaan sitä pitää virheellisenä. Luottamusvälien ulkopuolelle menevät arvot näkyvät jakaumissa pitkinä häntinä. Virheellisen arvon sisältämä tapaus voidaan joko poistaa tai tehdä virheelliselle arvolle jokin toimenpide, jota käytetään esimerkiksi puuttuville arvoille.

4. Neuroverkkojen muodostaminen aineistoille

Tässä luvussa opetetaan luvussa 3 esitetyt aineistot neuroverkoille. Neuroverkot opetetaan skaalaamalla aineistot lineaarisesti (kaava (3.1)) ja sen jälkeen käytetään muita luvussa 3 mainittuja menetelmiä aineistojen tarkempaan käsittelyyn.

Koska aineistot ovat neuroverkoille liian pieniä, käytetään ristiinvalidointia neuroverkkoja opettaessa. Ristiinvalidointi tehdään kymmenen kertaa (kymmenkertainen ristiinvalidointi) valiten jokaisella kierroksella testiaineistoon satunnaisesti kymmenen prosenttia koko aineistosta. Satunnaisella valinnalla testi- ja opetusaineistojen jakaumat tulevat olemaan lähes samat. Jokainen testijoukko on erillinen, eli kahden eri testijoukon leikkaus on tyhjä joukko. Kaikilla opetusjoukoilla opetetaan ja testijoukoilla testataan neuroverkkoja kymmenen kertaa, jotta mahdollinen globaali minimi voisi löytyä (kuva 13). Jokainen ristiinvalidointikerta opettaa neuroverkon yhteensä 100 kertaa.

Ristiinvalidoinnilla ei saada tiettyä neuroverkkoa, jota voidaan käyttää ongelman ratkaisemiseen, kuten kuvan 9 neuroverkko, vaan se antaa kuvan miten hyvin neuroverkot pystyvät oppimaan kyseisen aineiston. Jokaisella ristiinvalidointikerralla opetus- ja testiaineistot ovat erilaisia. Koska jokaisella ristiinvalidointikerralla opetus- ja testiaineistot eivät ole samoja, eri opetuksia ei voida suoraan verrata keskenään. Vertailemiseen käytetään testiaineistojen antamia keskiarvoja ja mediaaneja, joiden perusteella tehdään johtopäätöksiä neuroverkkojen oppimisesta. Ristiinvalidoinneissa käytettäviä opetus- ja testiaineistojen jakoja ei tallenneta, vaan ne ovat joka kerta uudet satunnaiset joukot. Eri ristiinvalidointikertojen vertaileminen onnistuu testiaineistoista saatujen keskiarvojen ja mediaanin perusteella, vaikka niiden testijoukkojen jako on eri.

Aineistojen pienten kokojen vuoksi neuroverkkojen opetuksessa ei ole voitu käyttää validointiaineistoa. Tätä varten neuroverkkojen opetuksia on täytynyt tarkastella vain testiaineistoista saatujen tulosten perusteella.

Jokaisella kierroksella tallennetaan neuroverkon testiaineistolle antama tarkkuus

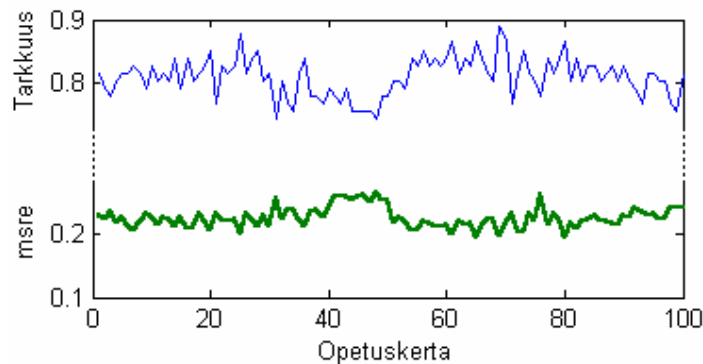
$$acc = 100 \frac{\sum_{c=1}^c tpos_c}{\sum_{c=1}^c pos_c} \% , \quad (4.1)$$

missä c on luokka, C on luokkien lukumäärä, $tpos_c$ on luokan c oikein luokiteltujen tapausten lukumäärä ja pos_c luokan c tapausten lukumäärä [Viikki, 2002]. Lisäksi tallennetaan myös *true positive rate* -arvot (jatkossa tpr)

$$tpr_c = 100 \frac{tpos_c}{pos_c} \% \quad (4.2)$$

kaikista luokista [Viikki, 2002]. Se kertoo, kuinka monta prosenttia luokkaan c kuuluvista tapauksista luokituu oikein. Tarkkuus kertoo, kuinka hyvin neuroverkko on pystynyt luokittelemaan aineiston ja tpr miten hyvin neuroverkko pystyy luokittelemaan eri luokkia.

Neuroverkkoja on opetettu erilaisilla opetusparametreilla ja arkkitehtuureilla. Opetusten tuloksia on verrattu toisiinsa vertailemalla niiden testiaineistoille antamien tarkkuuksien, tpr -arvojen ja keskineliövirheiden neliöjuurien keskiarvoja ($msre$) toisiinsa. Yleensä parhaan opetuksen testiaineistolle antaman keskineliövirheen neliöjuuren keskiarvo on pienin. Neuroverkon testiaineistosta antaman tarkkuuden keskiarvo kertoo tarkemmin miten hyvin neuroverkko osaa luokitella kyseistä aineistoa. Keskineliövirheen neliöjuuri ja tarkkuus ovat yleensä kääntäen verrannollisia ja ne täydentävät toisiaan. Kuvasta 25 näkee tarkkuuden yleensä olevan korkea, kun keskineliövirheen neliöjuuri on pieni. Yleensä keskineliövirheen neliöjuurella voidaan tarkentaa neuroverkon luokittelukykyä ja havaita helpommin neuroverkon mahdollinen ylioppiminen.



Kuva 25. Erään neuroverkon testiaineistosta antamien tarkkuuksien ja keskineliövirheiden neliöjuuren (paksu viiva) vertailua havainnollistava kuvaaja, josta huomaa niiden olevan yleisesti kääntäen verrannollisia.

Tämän luvun opetuksissa kokeillaan, kuinka hyvin neuroverkko pystyy luokittelemaan aineistoja yksinkertaisilla aineistojen käsittelyllä. Kukin aineiston attribuutti ja luokkamuuttuja skaalataan lineaarisesti välille $[0, 1]$ ja neuroneiden aktivaatiofunktioina käytetään logaritmistä sigmoidia, joka antaa arvoja väliltä $(0, 1)$. Aineistoja voidaan muokata muullakin tavalla

neuroverkkoja varten. Esimerkiksi luvussa 3 mainittuja frekvenssien tasaamista ei tehdä, sillä aineistot ovat pieniä. Muun muassa Ecoli-aineistossa jaetaan luokka viiden bitin jonoksi pienten luokkien poistamisen jälkeen.

Opetusalgoritmina käytetään *joustavaa takaisinlevitystä* (resilient back propagation). Joustava takaisinlevitys oli MATLAB:n [2002] testeissä keskimäärin paras opetusalgoritmi luokittelussa, mitä tässä tutkielmassa käytettävät aineistot ovat. Joskus painoarvoihin tehtävät muutokset ovat varsin pieniä, vaikka paikallinen minimi olisi kaukana niistä. Joustavan takaisinlevityksen tarkoituksena on välttää tätä ja siten nopeuttaa neuroverkon oppimista. Joustavassa takaisinlevityksessä on kaksi opetusparametriä korjauskertoimen muuttamista varten. Lisäksi on opetusparametri opetuksen lopettamiseksi, kun gradientti menee tarpeeksi pieneksi. Tästä käytetään nimitystä pienin gradientti. Se mahdollistaa, ettei neuroverkkoa opeteta turhaan, kun se ei enää suppene tarpeeksi. [MATLAB, 2002]

4.1. Alokas

Alokas-aineisto poikkeaa sekä attribuuttien että tapausten lukumäärän suhteen muista tässä tutkielmassa käytetyistä aineistoista. Tässä tutkielmassa käytetyssä alokas-aineistossa on peräti 10000 tapausta ja 39 attribuuttia. Alokas-aineiston suuruus on riittävä neuroverkolle, joten sille ei tarvitse tehdä ristiinvalidointia. Luokiteltavia muuttujia ovat normaali kuulo (luokka *normaali*) ja huono kuulo (luokka *huono*). Molemmilla luokilla on aineistossa 5000 tapausta, joten luokkamuuttuja on jakautunut tasaisesti. Alkuperäisessä 450 000 tapauksen aineistossa oli vain kaksi prosenttia huonokuuloisia. Neuroverkon olisi annettava vähintään 50 prosentin tarkkuus, jotta tulos olisi parempi kuin mitä saataisiin satunnaisesti arvaamalla jokainen tapaus. Opetusaineistoon on valittu satunnaisesti 67 prosenttia kaikista tapauksista ja loppuja 33 prosenttia tapauksista on käytetty testiaineistona. Opetusparametrit olivat jokaisessa opetuksessa 1000 epookkia, keskineliövirheen neliöjuurena 0,05, vahvistuskertoimenä 0,1 ja pienimpänä gradienttina 0,0002. Jokaisella opetuskerralla neuroverkkoja opetettiin kymmenen kertaa, jotta välttyttäisiin paikallisilta minimeiltä. Opettamiset aloitettiin yhdellä piiloneuronilla ja lisättiin piiloneuronien määrää vähitellen.

Ensiksi kokeiltiin opettaa neuroverkolle alokas-aineisto kaikilla 39 attribuutilla, vaikka aineistossa on muutama lähes muuttumaton attribuutti. Aluksi neuroverkkoon tuli siis 39 syötöneuronia. Kaksi luokkaa pystytään koodaamaan yhdellä tulosneuronilla, jossa toinen luokka on 1 ja toinen 0. Koska jokaisessa opetusaineistossa tulee olemaan 6700 tapausta koko aineiston 10000 tapauksesta, niin nyrkkisäännön mukaan piiloneuroneja voisi olla korkeintaan 17.

Neuroverkko pääsi keskimäärin 53 prosentin tarkkuuteen kun tpr_{normaali} oli keskimäärin 94 prosenttia ja tpr_{huono} 14 prosenttia. Testiaineiston antamat tarkkuudet ja keskineliövirheiden neliöjuuret eivät muuttuneet juuri yhtään piiloneuroneja lisättäessä, vaan pysyivät lähes samoina.

Seuraavaksi poistettiin aineistosta lähes muuttumattomat attribuutit. Attribuutti poistettiin, jos yli 9990 tapausta olivat samoja kyseisellä attribuutilla. Karsinnan jälkeen aineistoon jäi jäljelle 22 attribuuttia. 22 attribuutilla nyrkkisäännön mukaan neuroverkossa voisi olla korkeintaan 29 piiloneuronia. Karsinnan jälkeen neuroverkolle opetettiin aineisto samoilla opetusparametreillä kuin aikaisemmin. Tällä kerralla kymmenen opetusta kesti kolmasosan verrattuna 39 attribuutin opetukseen. Karsitullakin aineistolla neuroverkko pääsi keskimäärin 53 prosentin tarkkuuteen, kuten 39 attribuutin aineistolla. Keskineliövirheet ja tpr -arvot olivat myös samoja molemmilla eri aineistoilla.

Alokas-aineisto on selvästi vaikeasti opittava, eikä neuroverkko pysty oppimaan sitä. Kun tarkastellaan tpr -arvoja, voidaan päätellä neuroverkon lähes aina luokittelevan jokaisen tapauksen luokkaan *normaali*, sillä tpr_{normaali} on selvästi parempi kuin tpr_{huono} . Syötöneuronien vähentäminen, eli muuttumattomien attribuuttien poistaminen, vähensi selvästi neuroverkon opetusaikaa, mutta tarkkuuteen ja tpr -arvoihin se ei vaikuttanut. Aineiston oppimisen teki vaikeaksi mm. suuri samojen tapauksien lukumäärä eri luokilla. Esimerkiksi luokalla *normaali* oli 4572 ja luokalla *huono* oli 4214 tapausta joiden kaikki arvot olivat 0. Koska kahdella eri luokalla oli näin paljon samanlaisia tapauksia, niin verkko ei pystynyt oppimaan kumpaakaan luokkaa. Jos aineistosta karsi kaikki identtiset tapaukset pois oli aineiston koko enää vain 94, mikä on liian pieni neuroverkoille. Vaikka lähes muuttumattomat attribuutit poistettiin aineistosta, niin jäljelle jääneiden 22 attribuutin suhteelliset frekvenssit olivat 0,2-3,5 prosenttia eli kaikki attribuutit olivat selvästi vinosti jakautuneita, mikä mahdollisesti vaikutti myös neuroverkon oppimisvaikeuteen.

4.2. Buba

Buba-aineiston luokkamuuttujalla on kaksi luokkaa: *sairaat* ja *terveet*. Luokkamuuttujan luokkien frekvenssit ovat 145 (*sairaat*) ja 200 (*terveet*), joten tulokset ovat jakautuneet suhteellisen tasaisesti. Neuroverkon antaman tarkkuuden tulisi olla yli 58 prosenttia, jotta sen voitaisiin katsoa luokittelevan Buba-aineistoa. 58 prosentin tarkkuus vastaa tilannetta, jossa kaikkien tapausten arvataan kuuluvan suurimpaan luokkaan. Aluksi asetettiin neuroverkon tavoittelemaksi keskineliövirheeksi 0,025, laskettaviksi epookeiksi 10000, vahvistuskertoimeksi 0,1 ja pienimmäksi gradientiksi 0,0002.

Buba-aineistossa on kuusi attribuuttia, joten neuroverkkoon tulee kuusi syötöneuronia. Kaksi luokkaa pystytään koodaamaan yhdellä tulosneuronilla, jossa toinen luokka on 1 ja toinen 0. Koska jokaisessa opetusaineistossa tulee olemaan 311 tapausta koko aineiston 345 tapauksesta, niin nyrkkisäännön mukaan piiloneuroneja voisi olla korkeintaan neljä.

Opettaminen aloitettiin yhden piiloneuronin neuroverkosta. Jokaisen kerran jälkeen lisättiin yksi piiloneuroni, kunnes neuroverkon tarkkuus tai keskineliönvirheen neliöjuuri testiaineistoon nähden alkoi heiketä. Vielä varmistaakseen neuroverkon testiaineistolle antaman tarkkuuden ja keskineliövirheen neliöjuuren heikkenemisen, opetettiin neuroverkkoa viidellä piiloneuronilla. Ennen heikkenemistä olleen neuroverkon arkkitehtuurin katsottiin olevan paras Buba-aineiston luokittelamiseen.

Taulukko 2. Buba-aineiston opetuksen tulokset 10000 epookilla
(keskiarvo/mediaani).

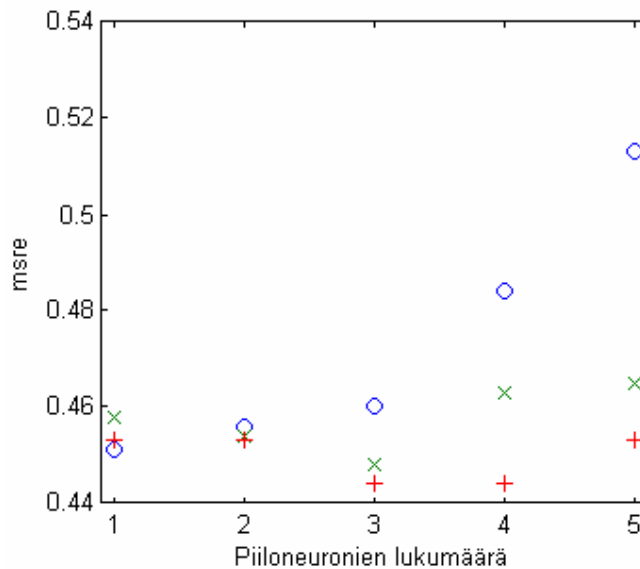
	Tarkkuus	tpr_{terve}	tpr_{sairas}
1 piiloneuroni	70%/68%	85%/88%	51%/50%
2 piiloneuronia	70%/71%	83%/83%	52%/52%
3 piiloneuronia	71%/71%	81%/81%	57%/57%
4 piiloneuronia	69%/71%	78%/78%	58%/57%
5 piiloneuronia	68%/68%	76%/76%	56%/56%

Taulukosta 2 voidaan huomata neuroverkon antaman tarkkuuden heikkeneminen kolmen piiloneuronin neuroverkon jälkeen. Tämän perusteella voidaan päätellä kolmen piiloneuronin neuroverkon olevan paras arkkitehtuuri Buba-aineistolle. Viiden piiloneuronin neuroverkon tuloksen heikkeneminen neljän piiloneuronin neuroverkkoon nähden vahvistaa tulosta.

Taulukko 3. Buba-aineiston opetuksen tulokset 1000 epookilla
(keskiarvo/mediaani).

	Tarkkuus	tpr_{terve}	tpr_{sairas}
1 piiloneuroni	70%/71%	85%/87%	47%/53%
2 piiloneuronia	71%/74%	82%/82%	56%/55%
3 piiloneuronia	72%/72%	81%/81%	59%/58%
4 piiloneuronia	70%/71%	79%/79%	60%/58%
5 piiloneuronia	71%/71%	79%/80%	59%/58%

Opettaessa neuroverkkoa 10000 epookkiin asti neuroverkon keskineliövirheen neliöjuuri suppeni lähes koko ajan hyvin vähän. Tätä varten kokeiltiin opettaa neuroverkkoa muuten samoilla opetusparametreillä, mutta vain 1000 epookkiin asti. Tämän opetuksen tulokset on koottu taulukkoon 3. Vertailemalla taulukoiden 2 ja 3 tuloksia huomataan neuroverkon suorituskyvyn olevan keskimäärin prosenttiyksikön parempi 1000 epookin opetuksessa kuin 10000 epookin. Kuvan 26 kuvaajasta voidaan nähdä kolmen opetuksen testiaineistoille antamien keskineliövirheiden neliöjuurien keskiarvojen erot, kun neuroverkkoa on opetettu 500, 1000 ja 10000 epookilla. Niistä huomaa selkeästi 10000 epookin opetuksen ylioppimisen keskineliövirheen neliöjuuren suurempina arvoina. Ero on sitä suurempi, mitä enemmän neuroverkossa on piiloneuroneja. 500 ja 1000 epookin opetuksissa ero on pieni, mutta voidaan päätellä 1000 epookin opetuksessa neuroverkon jo ylioppivan.



Kuva 26. Eri piiloneuronien määrillä neuroverkkojen testiaineistoille antamien keskineliövirheiden neliöjuurien keskiarvot 500 (+), 1000 (x) ja 10000 (o) epookin opetuksilla.

Paras arkkitehtuuri kolmen opetuksen perusteella Buba-aineistolle on kolmen piiloneuronin arkkitehtuuri. Tulosta vahvistaa myös kuvan 26 kuvaaja, josta voidaan huomata neuroverkon testiaineistolle antaman keskineliövirheen neliöjuuren keskiarvot eri piiloneuronien määrillä. Kuvaajasta voidaan huomata selkeästi keskineliövirheen neliöjuuren kasvavan kolmen piiloneuronin jälkeen. Keskineliövirheen neliöjuuren kasvaminen ja tarkkuuden huononeminen ovat merkkejä, että neuroverkko alkaa ylioppia aineistoa. Tässä neuroverkon testiaineistosta antamat keskineliövirheiden neliöjuuret ja tarkkuudet huononevat neljän piiloneuronin neuroverkolla. Voidaan tehdä johtopäätös, että kolmen piiloneuronin arkkitehtuuri on paras Buba-aineiston luokitteluun, eikä piiloneuroneja kannata lisätä enempää.

Kolmen piiloneuronin neuroverkon antama 71 prosentin tarkkuus on hyvä ja se ylittää asetetun 58 prosentin rajan. Tuloksista voidaan huomata kuinka pienikin ero luokkien frekvensseissä voi vaikuttaa neuroverkon oppimiseen. Arvojen tpr_{terve} ja tpr_{sairas} välinen ero on 20-30 prosenttiyksikköä, vaikka luokkien frekvenssien ero ei ole suuri. Lisäksi suurempi piiloneuronien määrä antaa paremman tpr_{sairas} -arvon.

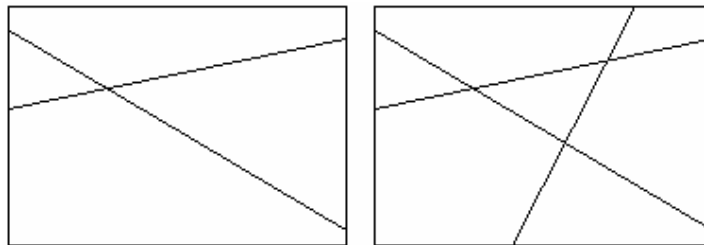
4.3. Ecoli

Ecoli-aineistossa on eniten luokkia tässä tutkielmassa käytetyistä kahdeksasta aineistosta. Mutta kuten luvussa 3 mainittiin, niistä osan frekvenssi on liian pieni neuroverkoille ja ne joudutaan poistamaan. Opetuksiin poistettiin luokat

omL, *imL* ja *imS*, joiden frekvenssit ovat viisi, kaksi ja kaksi. Näiden luokkien poistamisen jälkeen *ecoli*-aineistossa on huimaus-aineiston jälkeen eniten luokkia. Lisäksi aineistosta poistettiin attribuutit *chg* ja *lip*, sillä em. luokkien poistamisen jälkeen ne olivat melkein vakiomuuttujia. Liian pienten luokkien poistamisen jälkeen aineistossa on viisi luokkaa: *cp*, *im*, *pp*, *imU* ja *om*. Luokkien frekvenssit ovat 143 (*cp*), 77 (*im*), 52 (*pp*), 35 (*imU*) ja 20 (*om*). Neuroverkon on annettava vähintään 43 prosentin tarkkuudella oikea luokka, jotta sen voitaisiin sanoa luokittelevan aineistoa oikein (vrt. kaikkien tapausten arvaaminen suurimpaan luokkaan).

Ecoli-aineistoon jää viisi attribuuttia *chg*- ja *lip*-attribuuttien poistamisen jälkeen, joten neuroverkkoon tulee viisi syötöneuronia. Kaksi luokkaa pystytään koodaamaan yhdellä tulosneuronilla, jossa toinen luokka on 1 ja toinen 0. Viiden luokan tapauksessa on luontevinta koodata luokat viitenä tulosneuronina, jossa kukin tulosneuroni vastaa omaa luokkaa. Kukin tulosneuroni kertoo, kuinka paljon annettu tapaus muistuttaa kutakin luokkaa. Tällöin neuroverkko sanoo kyseisen tapauksen muistuttavan eniten tätä luokkaa, jota vastaavan tulosneuronin tulos on suurin. Koska jokaisessa opetusaineistossa tulee olemaan pienten luokkien poistamisen jälkeen 294 tapausta koko aineiston 327 tapauksesta ja syöte- ja tulosneuroneja on viisi, niin nyrkkisäännön mukaan piiloneuroneja voisi neuroverkossa olla korkeintaan kolme aineiston oppimiseen.

Koska *ecoli*-aineistossa on karsinnan jälkeen viisi luokkaa, on neuroverkossa oltava vähintään kolme piiloneuronia luokitellakseen kaikki viisi luokkaa. Luvussa 2 mainittiin, että neuroverkko pystyy rajoittamaan tasolta piiloneuronien lukumäärien mukaisten suorien määrittämien alueiden määrän luokkia. Kuva 27 havainnollistaa, miksi neuroverkossa tarvitaan vähintään kolme piiloneuronia luokitellakseen viisi luokkaa. Kahdella suoralla pystytään erottamaan korkeintaan neljä luokkaa, kun kolmella suoralla pystytään rajoittamaan seitsemän luokkaa.



Kuva 27. Vasemmalla on esimerkki kahden ja oikealla kolmen suoran mahdollistamista alueiden rajaamisesta. Kahdella suoralla pystytään rajaamaan korkeintaan neljä ja kolmella seitsemän aluetta.

Neuroverkon opettaminen aloitettiin kahdella piiloneuronilla, vaikka se ei periaatteessa riitä viiden luokan luokitteluun. Tämän jälkeen lisättiin piiloneuronien lukumäärää vähitellen, kunnes neuroverkon testiaineistoista antamien keskineliövirheiden neliöjuurien keskiarvo alkoi nousta tai ei muuttunut enää paljoa. Neuroverkkoa kokeiltiin opettaa erilaisilla opetusparametreillä, pitäen vahvistuskertoimen jokaisessa opetuksessa 0,1:nä ja pienimmän gradientin 0,0002:na. Muut opetuksissa käytetyt parametrit ovat taulukossa 4.

Taulukko 4. Ecoli-aineiston opetusparametrejä.

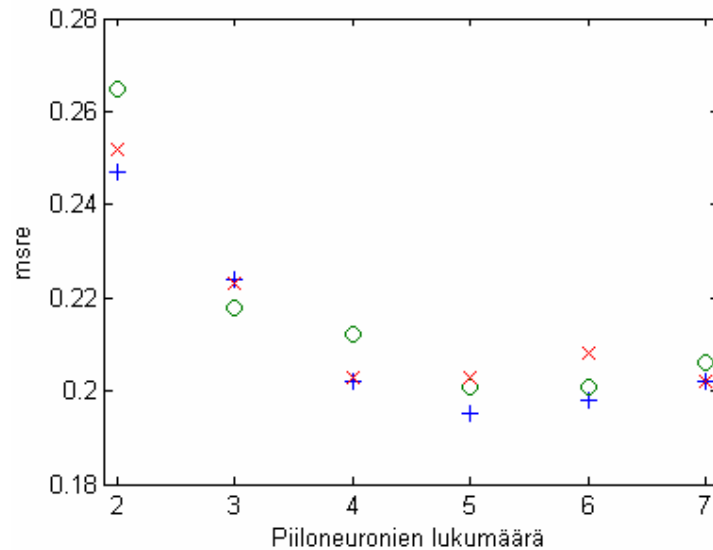
	Epookit	Virhe
Opetus 1	1000	0,025
Opetus 2	500	0,025
Opetus 3	2000	0,01

Jo opetuksessa 1 huomattiin neuroverkon kykenevän luokitteluun ecoli-aineistoa hyvin. Taulukossa 5 olevista tuloksista näkee neuroverkon pystyvän keskimäärin 87 prosentin tarkkuuteen opetuksella 1. Opetuksilla 2 ja 3 neuroverkon antamat arvot eivät muuttuneet paljoa opetuksesta 1. Ecoli-aineistonkin tapauksessa neuroverkko oppi parhaiten suurimman luokan. Tällöin tpr_{cp} on koko ajan lähellä 100 prosenttia ja mediaani on 100 prosenttia. Yllättäen luokan om , jolla on pienin frekvenssi, luokittelu onnistui hyvin tpr_{om} -arvon saavutettua jatkuvasti yli 80 prosentin arvoja.

Taulukko 5. Ecoli-aineiston opetuksen tuloksia eri arkkitehtuureilla opetuksella 1 (keskiarvo/mediaani).

	Tarkkuus	tpr_{om}	tpr_{imU}	tpr_{pp}	tpr_{im}	tpr_{cp}
2 piiloneuronia	77%/81%	20%/0%	26%/20%	75%/83%	84%/89%	95%/100%
3 piiloneuronia	83%/84%	60%/75%	48%/50%	83%/83%	79%/83%	96%/100%
4 piiloneuronia	88%/88%	81%/100%	60%/67%	81%/100%	84%/83%	98%/100%
5 piiloneuronia	87%/88%	81%/100%	68%/71%	89%/100%	81%/80%	97%/100%
6 piiloneuronia	88%/88%	88%/100%	59%/67%	84%/100%	82%/86%	98%/100%
7 piiloneuronia	87%/88%	76%/100%	64%/67%	87%/95%	80%/80%	97%/100%

Taulukon 5 sarakkeen tarkkuus perusteella neuroverkolle optimaalinen määrä piiloneuroneja on neljä, viisi, kuusi tai seitsemän. Koska eri opetuksien antamat tarkkuudet ja tpr -arvot olivat lähes samoja, on opetuksien välisiä eroja tarkasteltava neuroverkon testiaineistoista antamien keskineliövirheiden neliöjuurien keskiarvojen perusteella. Kuvan 28 kuvaajasta voidaan nähdä opetuksien olevan lähes yhtä hyviä neljästä piiloneuronista lähtien. Missään kolmesta opetuksesta neuroverkko ei ala selvästi ylioppimaan aineistoa millään neuroverkon arkkitehtuurilla. Opetus 1 on vähän parempi opetuksia 2 ja 3, josta voidaan päätellä opetuksien 2 ja 3 ylioppineen aineistoa. Erot näkyvät selkeimmin yli neljän piiloneuronin arkkitehtuurin neuroverkoissa. Viiden piiloneuronin jälkeen keskineliövirheiden neliöjuurien keskiarvo ei muutu merkittävästi, jolloin viittä piiloneuronia voidaan pitää ecoli-aineiston luokittelamiseen parhaana arkkitehtuurina. Paras tulos viiden piiloneuronin arkkitehtuurilla pystytään saavuttamaan opetuksen 1 opetusparametreilla.



Kuva 28. Opetusten 1 (+), 2 (o) ja 3 (x) testiaineistoista antamien keskineliövirheiden neliöjuurien keskiarvot eri piiloneuronien määrillä.

Ecoli-aineiston luokittelussa pystytään huomaamaan miten piiloneuronien määrä vaikuttaa neuroverkon luokittelmien luokkien määrään. Kahdella piiloneuronilla (taulukko 5) neuroverkko pystyi luokittelemaan hyvin luokat *cp*, *im* ja *pp*, mutta luokkien *imU* ja *om* luokittelu oli vaikeata. Neuroverkko ei kyennyt luokittelemaan viittä luokkaa kahdella piiloneuronilla, koska neuroverkko pystyy kahdella piiloneuronilla erottamaan korkeintaan neljä aluetta tasosta (kuva 27). Pienillä piiloneuronien määrällä neuroverkko pystyi luokittelemaan ainoastaan suurempia luokkia ja pienemmät luokat jäivät luokittelematta. Taulukosta 5 ja kuvasta 28 voidaan nähdä selvät erot kahden, kolmen ja neljän piiloneuronin arkkitehtuureissa. Varsinkin kahden piiloneuronin arkkitehtuurilla keskineliövirheen neliöjuuri on selvästi suurempi kuin yli kahden piiloneuronin arkkitehtuureilla.

4.4. Haberman

Haberman-aineistossa on kaksi luokkaa: potilaat, jotka selviytyivät viisi vuotta tai enemmän leikkauksesta (luokka *selviytyi*) ja potilaat, jotka kuolivat viiden vuoden sisällä leikkauksesta (luokka *kuoli*). Luokkamuuttujan frekvenssit ovat 225 (*selviytyi*) ja 81 (*kuoli*), eli se ei ole jakautunut tasaisesti. Neuroverkon on annettava vähintään 73 prosentin tarkkuudella oikea luokka, jotta sen voitaisiin sanoa luokittelleen aineistoa oikein (vrt. kaikkien tapausten arvaaminen suurimpaan luokkaan). Opetusparametreiksi jokaisella opetuksella asetettiin 1000 epookkia, 0,025 keskineliövirheeksi, 0,1 korjauskertoimeksi ja 0,0002 pienimmäksi gradientiksi.

Haberman-aineistossa on kolme attribuuttia, joten neuroverkkoon tulee kolme syötöneuronia. Kaksi luokkaa pystytään koodaamaan yhdellä tulosneuronilla, jossa toinen luokka on 1 ja toinen 0. Koska jokaisessa opetusaineistossa tulee olemaan 275 tapausta koko aineiston 306 tapauksesta, niin nyrkkisäännön mukaan piiloneuroneja voisi olla korkeintaan seitsemän.

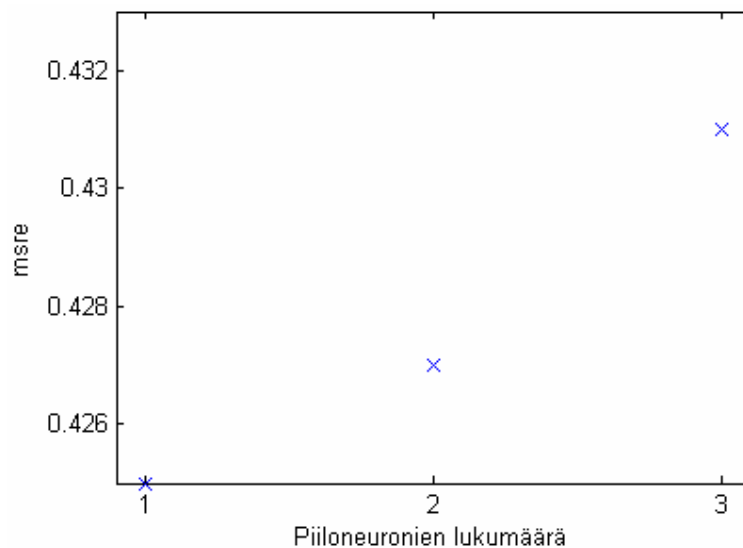
Haberman-aineistossa on tässä tutkielmassa käytetyistä aineistoista vähiten attribuutteja, kolme. Siksi kokeiltiin kaikkia yhden piilokerroksen arkkitehtuuria, jotka täyttävät Swinglerin [1996] ehdon: piirteiden poistamiseen tarvitaan vähemmän piiloneuroneja kuin syötöneuroneja. Yhden piilokerroksen neuroverkkoa kokeiltiin yhdellä, kahdella ja kolmella piiloneuronilla. Suurempia määriä piiloneuroneja on turha kokeilla em. ehdon perusteella. Taulukon 6 tuloksista voidaan päätellä jokaisen arkkitehtuurin olevan lähes samantasoisia. Kaikki kolme arkkitehtuuria pääsevät lähelle Haberman-aineistolle asetettua 73 prosentin rajaa, joten ne pystyvät luokittelemaan huonosti Haberman-aineistoa, sillä arvaamalla kaikki tapaukset suurimpaan luokkaan päästään keskimäärin samoihin tarkkuuksiin. Viisi vuotta tai enemmän selviytyneiden potilaiden (*selviytyi*) luokittelu onnistuu neuroverkolta kiitettävästi, mutta viiden vuoden sisällä kuolleiden potilaiden (*kuoli*) luokittelu onnistuu huonosti. Tämän perusteella neuroverkko luokittelee suurimman osan tapauksista luokkaan *selviytyi*.

Taulukko 6. Haberman-aineiston opetuksen tuloksia kolmella eri arkkitehtuurilla (keskiarvo/mediaani).

	Tarkkuus	tpr_{kuoli}	$tpr_{selviytyi}$
1 piiloneuroni	73%/73%	32%/29%	87%/87%
2 piiloneuronia	73%/70%	29%/31%	88%/88%
3 piiloneuronia	73%/70%	31%/33%	88%/90%

Haberman-aineiston opettaminen tapahtui nopeasti, sillä neuroverkon opettamiseen meni keskimäärin vain 1000 epookkia. Neuroverkon keskineliövirheen neliöjuuren suppeneminen meni nopeasti alle pienimmän gradientin, joka oli 0,0002. Vähäinen attribuuttien ja tapausten määrä vei neuroverkon nopeasti lähelle optimaalista tulosta ja opettaminen ei kestänyt kauaa.

Tässä kokeilluista arkkitehtuureista mitään ei voida pitää muita selvästi parempana arkkitehtuurina Haberman-aineistolle taulukon 6 perusteella. Kaikkien arkkitehtuurien antamat tarkkuudet ovat melkein samat ja ne pääsevät keskimäärin vain 73 prosentin tarkkuuteen. Kuvassa 29 on piirretty neuroverkon testiaineistoille antamat keskineliövirheiden neliöjuurien keskiarvot tässä testatuilla arkkitehtuureilla. Voidaan huomata yhden piiloneuronin arkkitehtuurin antaman keskineliövirheen neliöjuuren olevan keskimäärin parempi (pienempi) kuin kahden ja kolmen piiloneuronin, kuten oli myös tarkkuuksissa taulukossa 6. Erot ovat niin pieniä, ettei mitään lopullista ratkaisua parhaasta arkkitehtuurista Haberman-aineistoon voida tällä perusteella tehdä. Voidaan huomata neuroverkon alkavan vähitellen ylioppia aineistoa piiloneuronien määrän kasvaessa.



Kuva 29. Yhden kahden ja kolmen piiloneuronin arkkitehtuurien testiaineistolle antamien keskineliövirheiden neliöjuurien keskiarvot.

Haberman-aineiston opetuksen tuloksista voidaan huomata, kuinka paljon luokkien frekvenssien ero vaikuttaa neuroverkon oppimiseen. Viisi vuotta tai enemmän selviytyneiden potilaiden (*selviytyi*) *tpr* on huomattavasti suurempi kuin alle viisi vuotta selvinneiden potilaiden (*kuoli*) *tpr*. Voidaan huomata kuinka paljon paremmin neuroverkko oppii suurimmat luokat. Haberman-aineiston tapauksessa *tpr*-arvojen ero on keskimäärin yli 50 prosenttiyksikköä kun Buba-aineistossa se oli 20-30 prosenttiyksikköä.

4.5. Huimaus

Huimaus-aineistossa on 815 tapausta ja kuusi luokkaa: *an*, *bpv*, *Meniere*, *sd*, *tv* ja *vn*. Luokkien frekvenssit ovat 130 (*an*), 146 (*bpv*), 313 (*Meniere*), 41 (*sd*), 65 (*tv*) ja 120 (*vn*). Luokkien jakauma ei siis ole tasainen, vaikka pienimmät luokat on

poistettu tässä tutkielmassa käytetystä aineistosta. Neuroverkon on annettava vähintään 38 prosentin tarkkuudella oikea luokka, jotta sen voitaisiin sanoa luokittelevan aineistoa oikein (vrt. kaikkien tapausten arvaaminen suurimpaan luokkaan). Alkuperäisessä huimaus-aineistossa on yhteensä 40 attribuuttia, joista tässä tutkielmassa on käytetty aluksi 13 merkittävintä ja sen jälkeen yhdeksää merkittävintä attribuuttia. Attribuuttien määrän karsimiseen on ollut syynä mm. se, että neuroverkon painoarvojen määrä saataisiin pidettyä pienempänä ja tällä tavalla neuroverkko pystyisi vielä oppimaan aineiston. 40 attribuuttia olisi liian suuri määrä neuroverkoilla, kun tapauksia on vain 815.

Huimaus-aineistossa luokkamuuttuja on koodattu samalla tavalla kuin *ecoli*-aineistossa, missä kustakin luokasta on tehty oma tulosneuron. Kukin tulosneuron kertoo, kuinka paljon tapaus muistuttaa kyseistä luokkaa. Neuroverkon katsotaan luokittelevan tapauksen siihen luokkaan, jonka tulosneuronin antama tulos on suurin. Koska jokaisessa opetusaineistossa tulee olemaan 734 tapausta ja syötoneuroneja on 13 ja tulosneuroneja kuusi, niin nyrkkisäännön mukaan piiloneuroneja voisi neuroverkossa olla korkeintaan neljä aineiston oppimiseen.

Kuuden luokan luokitteluun neuroverkossa on oltava vähintään kolme piiloneuronia, koska neuroverkko pystyy rajoittamaan tasolta piiloneuronien lukumäärien mukaisten suorien määrittämien alueiden määrän luokkia. Kuutta luokkaa varten tarvitaan vähintään kolme suoraa (kuva 27).

Neuroverkon opettaminen aloitettiin kahdella piiloneuronilla, vaikka se ei riitä huimaus-aineistossa olevan kuuden luokan luokitteluun. Tällä tavalla pystytään havainnollistamaan, miten piiloneuronien lukumäärä vaikuttaa luokiteltavien luokkien määrään. Tämän jälkeen lisättiin piiloneuronien määrää vähitellen, kunnes neuroverkon testiaineistosta antamien keskineliövirheiden neliöjuurien keskiarvo alkoi nousta tai ei muuttunut enää paljoa. Neuroverkkoa kokeiltiin opettaa erilaisilla opetusparametreilla pitäen vahvistuskertoimen jokaisessa opetuksessa 0,1:nä ja pienimmän gradientin 0,0002:na. Muut opetuksissa käytetyt parametrit ovat taulukossa 7.

Taulukko 7. Huimaus-aineiston opetusparametrejä.

	Epookit	Virhe
Opetus 1	500	0,05
Opetus 2	500	0,01
Opetus 3	1000	0,01

Kuten *ecoli*-aineistonkin tapauksessa, ei huimaus-aineistoon luokitteleminen onnistunut kovin hyvin kahdella piiloneuronilla. Ero kahden ja kolmen piiloneuronin neuroverkkojen tarkkuudella oli noin viisi prosenttiyksikköä. Tarkastelemalla opetuksen 1 tuloksia taulukosta 8 voidaan huomata, että neljän piiloneuronin jälkeen eivät tarkkuudet ja *tpr*-arvot muutu kovin paljoa. Etenkin luokan *sd* luokitteleminen on selvästi vaikeaa neuroverkolle opetuksella 1.

Taulukko 8. Huimaus-aineiston opetuksen tuloksia eri arkkitehtuureilla opetuksella 1 (keskiarvo/mediaani).

	Tarkkuus	<i>tpr</i> _{an}	<i>tpr</i> _{tv}	<i>tpr</i> _{sd}	<i>tpr</i> _{Meniere}	<i>tpr</i> _{bpv}	<i>tpr</i> _{vn}
2 piiloneuronia	73%/73%	72%/80%	46%/40%	8%/0%	90%/91%	63%/67%	74%/78%
3 piiloneuronia	78%/78%	83%/82%	67%/75%	4%/0%	90%/91%	68%/69%	81%/85%
4 piiloneuronia	78%/79%	84%/85%	73%/83%	5%/0%	90%/90%	68%/71%	83%/85%
5 piiloneuronia	79%/79%	85%/87%	73%/80%	6%/0%	90%/90%	69%/69%	84%/85%
6 piiloneuronia	78%/80%	87%/90%	75%/80%	5%/0%	90%/90%	67%/69%	83%/83%
7 piiloneuronia	79%/80%	86%/86%	80%/83%	2%/0%	90%/90%	68%/69%	83%/84%

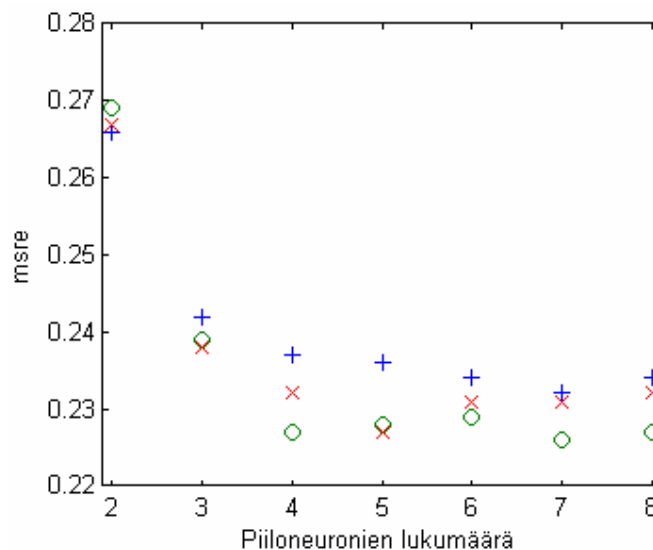
Opetuksella 1 opettaessa neuroverkko saavutti asetetun virheen jo alle 100 epookissa kun piiloneuronien määrä oli suuri, siksi opetuksen 2 tavoiteltu virhe asetettiin pienemmäksi. Tällä tavalla neuroverkkoa opetetaan enemmän. Opetuksen 2 antamat tarkkuudet olivat vähän parempia kuin opetuksen 1 antamat tarkkuudet. Taulukkoon 9 on kerätty opetuksen 2 antamat tarkkuudet ja tpr -arvot eri piiloneuronien määrillä. Etenkin tpr_{sd} on suuremmilla piiloneuronien määrillä selvästi parempi opetuksessa 2 kuin opetuksessa 1. Tästä voidaan päätellä, että luokan sd erottaminen aineistosta vaatii monimutkaisia neuroverkkoja. Selvästi luokan sd pieni koko aineistossa vaikuttaa siihen, miten neuroverkko pystyy oppimaan sen. Muidenkin luokkien tpr -arvot ovat suurempia tai lähes samoja opetuksessa 2 kuin opetuksessa 1. Tarkkuudet ovat keskimäärin pari prosenttiyksikköä paremmat opetuksessa 2 kuin opetuksessa 1.

Taulukko 9. Huimaus-aineiston opetuksen tuloksia eri arkkitehtuureilla opetuksella 2 (keskiarvo/mediaani).

	Tarkkuus	tpr_{an}	tpr_{tv}	tpr_{sd}	$tpr_{Meniere}$	tpr_{bpv}	tpr_{vn}
2 piilo-neuronia	72%/73%	77%/83%	41%/38%	6%/0%	89%/90%	66%/67%	70%/75%
3 piilo-neuronia	78%/79%	83%/85%	75%/80%	6%/0%	89%/90%	68%/68%	82%/83%
4 piilo-neuronia	80%/80%	89%/90%	79%/83%	11%/0%	89%/89%	69%/73%	84%/86%
5 piilo-neuronia	80%/80%	90%/91%	74%/78%	20%/0%	89%/89%	70%/69%	83%/85%
6 piilo-neuronia	80%/80%	88%/89%	78%/80%	20%/20%	89%/89%	72%/75%	84%/82%
7 piilo-neuronia	81%/81%	91%/91%	73%/75%	21%/25%	89%/89%	73%/75%	84%/85%

Opetuksen 3 tuloksen olivat lähes samat kuin opetuksen 2 tulokset (taulukko 9). Tästä voidaan päätellä, ettei neuroverkko pystyisi oppimaan huimaus-aineistoa ainakaan paljoa paremmin, kuin mitä se pystyy oppimaan opetuksella 2.

Taulukkojen 8 ja 9 perusteella optimaalinen määrä piiloneuroneja huimaus-aineiston luokitteluun olisi seitsemän piiloneuronia, koska silloin opetuksen 2 antama tarkkuus on ollut korkein. Tarkkuuksien erot ovat niin pienet, ettei niiden perusteella voida lopullista päätöstä tehdä. Tarkastellaan vielä tarkemmin opetusten eroja neuroverkkojen testiaineistoista antamien keskineliövirheiden neliöjuurien keskiarvojen perusteella (kuva 30). Tämä tulos vahvistaa jo tarkkuuksien ja piiloneuronien tasolle vetämien suorien (kuva 27) antamaa tulosta, että kuuden luokan luokitteluun tarvitaan enemmän kuin kaksi piiloneuronia. Kuvan 30 perusteella opetus 2 on paras, koska sillä opetettujen neuroverkkojen testiaineistoista antamien keskineliövirheiden neliöjuurien keskiarvot ovat keskimäärin pienempiä kuin opetuksilla 1 ja 3. Tarkastelemalla eri opetusten testiaineistoista antamien keskineliövirheiden neliöjuurien keskiarvoja eri piiloneuronien määrillä voidaan huomata, etteivät keskineliövirheiden neliöjuurien keskiarvot laske paljoa neljän piiloneuronin jälkeen. Koska myös eri opetusten antamat tarkkuudet eivät kasva paljoa neljän piiloneuronin jälkeen, voidaan tehdä johtopäätös, että neljä piiloneuronia olisi optimaalinen määrä piiloneuroneja huimaus-aineiston luokitteluun kun luokkien määrä on kuusi.



Kuva 30. Opetusten 1 (+), 2 (o) ja 3 (x) testiaineistoista antamien keskineliövirheiden neliöjuurien keskiarvot eri piiloneuronien määrillä.

Koska jokaisessa aiemmassa opetuksessa tpr_{sd} oli hyvin pieni, niin kokeiltiin opettaa neuroverkolle huimaus-aineistoa, josta on poistettu luokkaan sd kuuluvat tapaukset. On kuitenkin huomattava, että aikaisempien huimaus-aineiston opetusten tarkkuudet paranevat, kun ei oteta huomioon luokan sd tapauksia, jotka lähes kaikki ovat luokittuneet väärin. Tätä varten luokittelun parantumisen pitäisi olla itsestään selvää. Nyt onkin kiinnostavaa katsoa,

paraneeko tarkkuus enemmän, kuin mitä sen pitäisi parantua, jos luokan *sd* tapaukset olisi otettu pois aiempien opetusten testaamisesta.

Lasketaan nyt, kuinka paljon tarkkuuden pitäisi parantua luonnollisesti luokan *sd* tapausten poistamisen jälkeen. Johdetaan kaava tarkkuuden paranemiselle, kun testiaineistosta poistetaan luokka *x*, jonka *tpr* on 0 tai lähes 0. Käytössä olevat muuttujat ja niiden selitykset on kuvattu taulukossa 10. Koska *tpr_x* on lähes 0, niin luokan *x* tapausten poistaminen tarkkuuden laskemisesta nostaa tarkkuutta. Kun tästä vähennetään alkuperäinen tarkkuus, niin saadaan

$$acc' = \frac{n \cdot acc}{n - n_x} - acc = \frac{n \cdot acc - acc(n - n_x)}{n - n_x} = \frac{acc \cdot n_x}{n - n_x}, \quad (4.3)$$

mikä on tarkkuuden kasvu. Kun sijoitetaan tähän $acc = 80\%$, $n = 81$ ja $n_{sd} = 4$, saadaan tulokseksi noin neljä prosenttia. Jos tarkkuus paranee vähemmän kuin neljä prosenttiyksikköä luokan *sd* poistamisen jälkeen, niin luokan *sd* poistaminen ei paranna neuroverkon luokittelutarkkuutta, koska luokan *sd* poistaminen tarkkuuden laskemisesta parantaa tarkkuutta noin neljä prosenttiyksikköä.

Taulukko 10. Muuttujat ja niiden selitykset tarkkuuden paranemisen laskemiseen.

Muuttuja	Selitys
acc'	Tarkkuuden parannus luokan <i>x</i> poistamisen jälkeen.
acc	Kaikilla luokilla saatu tarkkuus.
n	Testiaineiston koko kaikilla luokilla.
n_x	Poistettavan luokan (<i>x</i>) tapausten lukumäärä testiaineistossa.

Luokan *sd* poistaminen huimaus-aineistosta paransi tarkkuutta vähän, mutta ei merkittävän paljoa. Tarkkuus parani keskimäärin 4-5 prosenttiyksikköä, joka on suurin piirtein sama kuin edellä laskettu vähimmäisparannus tarkkuuteen, joka oli noin neljä prosenttiyksikköä. Tällöin *tpr*-arvot eivät muuttuneet paljoa verrattuna opetuksiin, joissa oli mukana kaikki kuusi luokkaa. Tämän perusteella luokan *sd* poistamisella ei ole paljoa merkitystä muiden luokkien luokitteluun, mutta pieni parannus tarkkuuteen saadaan poistamalla luokan *sd* tapaukset huimaus-aineistosta.

Seuraavaksi vähennettiin attribuuttien lukumäärä yhdeksään, kun se aikaisemmissa opetuksissa oli 13. Attribuuttien vähentäminen antaa neuroverkolle nyrkkisäännön perusteella paremman mahdollisuuden oppia aineiston, koska tällöin piiloneuroneja pystytään käyttämään enemmän. Lisäksi huonommat attribuutit saattavat vain heikentää neuroverkon oppimista. Attribuuttien vähentäminen myös nopeuttaa yleensä neuroverkon oppimista.

Opetuksella 1 yhdeksän attribuutin aineistolla tarkkuus oli sama kuin 13 attribuutin aineistolla (taulukko 8) vaihdellen 76 ja 78 prosentin välillä piiloneuronien lukumäärästä riippuen. Opetuksella 1 yhdeksän attribuutin aineisto on parempi kuin 13 attribuutin aineisto, koska se on yksinkertaisempi ja täten mahdollistaa neuroverkon paremman ja nopeamman oppimisen. Opetuksilla 2 ja 3 taas yhdeksän attribuutin aineisto jopa heikensi tarkkuutta pari prosenttiyksikköä tarkkuuden ollessa 77 ja 80 prosentin välillä piiloneuronien lukumäärästä riippuen. Jokaisella kolmella opetuksella kunkin luokan *tpr*-arvot olivat lähes samat kuin aikaisemmissa opetuksissa. Tarkkuuden heikkeneminen yleisesti kaikilla kolmella opetuksella on niin pientä, että on syytä harkita attribuuttien määrän vähentämistä 13:sta yhdeksään. Parin prosenttiyksikön heikkeneminen tarkkuudessa ei välttämättä ole niin huono asia kuin miltä se näyttää. Yhdeksällä attribuutilla neuroverkolla on paremmat lähtökohdat oppia ja yleistää huimaus-aineistoa kuin 13 attribuutilla.

Huimaus-aineiston opettamisesta saatujen tulosten perusteella voidaan päätellä, että neuroverkko pystyy oppimaan huimaus-aineiston keskimäärin 80 prosentin tarkkuudella (taulukko 9). Tarkkuutta saatiin nostettua vähän kun poistettiin kaikki luokan *sd* tapaukset. Tällöin tarkkuudeksi saatiin keskimäärin 84 prosenttia. Sekä 80 että 84 prosentin tarkkuudet ylittävät asetetun 38 prosentin rajan, joten neuroverkon voidaan sanoa luokittelevan huimaus-aineistoa hyvin. Mikäli neuroverkon antamaa tarkkuutta halutaan maksimoida, niin opetettavassa aineistossa tulisi olla 13 tärkeintä attribuuttia huimaus-aineistosta. Toisaalta yhdeksällä attribuutilla tarkkuus ei ollut paljoa huonompi kuin 13 attribuutilla ja pienemmällä attribuuttien määrällä neuroverkolla on paremmat lähtökohdat oppia ja yleistää aineistoa. Taulukon 9 ja kuvan 30 perusteella voidaan päätellä, että neljän piiloneuronin neuroverkko olisi paras huimaus-aineiston luokitteluun. Huimaus-aineistonkin kohdalla neuroverkko oppi parhaiten suurimmat luokat ja pienintä luokkaa (*sd*) neuroverkko ei pystynyt luokittelemaan juuri yhtään. Käytännössä lähes kaikki luokaksi *sd* luokituttavat tapaukset olivat virheellisiä. Tämän takia luokkaan *sd* luokituttavia tapauksia ei kannata huomioida ollenkaan, sillä on hyvin

epätodennäköistä että ko. tapaus kuuluisi oikeasti luokkaan sd (ks. taulukko 9, tpr_{sd}).

4.6. New-thyroid

New-thyroid -aineistossa on vain 215 tapausta, mikä on pieni aineisto neuroverkkoja varten. Tämä rajoittaa nyrkkisäännön mukaan piiloneuronien lukumäärän kolmeen piiloneuroniin tulosneuronien määrän ollessa yksi ja syötoneuronien määrän ollessa viisi. New-thyroid -aineistossa on kolme luokkaa: *normaali*, *hyper* ja *hypo*. Linearisella skaalauksella luokkien arvoiksi saadaan 0, 0,5 ja 1,0. Luokkien frekvenssit ovat 150 (*normaali*), 35 (*hyper*) ja 30 (*hypo*), joten ne eivät ole jakautuneet tasaisesti. New-thyroid -aineiston tapauksessa neuroverkon antaman tarkkuuden tulisi olla yli 70 prosenttia, sillä muuten tulos on sama kuin mitä saadaan, jos arvataan jokainen tapaus suurimpaan luokkaan.

New-thyroid -aineistolla neuroverkon tulosten tulkitseminen ei ole niin yksinkertaista kuin esim. Buba- tai Haberman-aineistolla, joissa on vain kaksi luokkaa. Kahdella luokalla tuloksen voidaan jakaa helposti kahteen osaan siten, että toinen luokista on 0 ja toinen 1. Yleinen tapa useamman kuin kahden luokan tapauksissa on tehdä jokaisesta luokasta oma tulosneuron, mutta tässä on kokeiltu opettaa neuroverkkoa käsittelemällä aineistoa vain lineaarisella skaalauksella ja yhdellä tulosneuronilla. Tässä tapauksessa tulosneuronin arvoväli $[0, 1]$ on jaettu kolmeen osaan: $[0, 0,33]$, $(0,33, 0,67)$ ja $[0,67, 1,00]$. Tällöin jokaiselle luokalle tulee yhtä suuri alue tulosneuronin mahdollisista tuloksista.

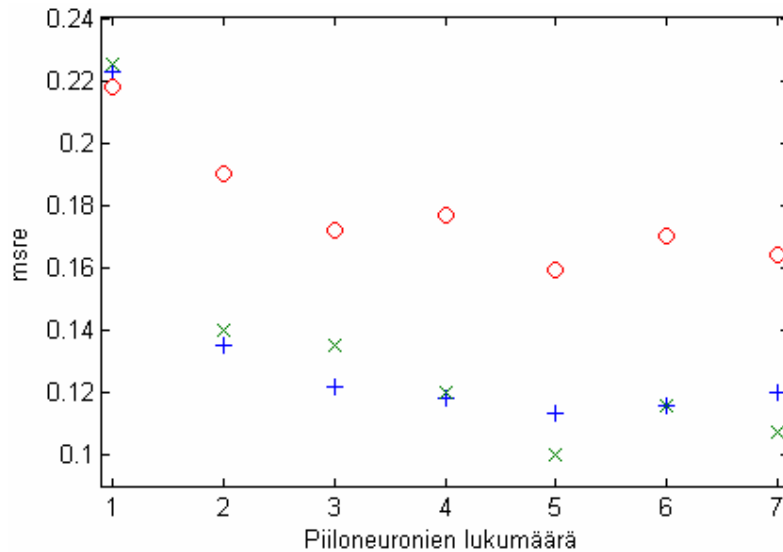
New-thyroid -aineiston luokittelu oli selvästi neuroverkolle helpointa tässä tutkielmassa käytetyistä kahdeksasta aineistosta. Muilla aineistoilla opetuksen keskineliövirheen neliöjuuri laski harvoin alle 0,1, kun new-thyroid-aineistolla se laski melkein aina alle 0,01. Neuroverkkojen antamat tarkkuudetkin olivat selvästi parempia new-thyroid -aineistolle kuin muille aineistoille.

Opetuksia tehtiin taulukossa 11 näkyvillä opetusparametreilla ja lisättiin piiloneuronien lukumäärää, kunnes neuroverkkojen testiaineistoille antamien keskineliövirheiden neliöjuurien keskiarvo alkoi kasvaa. Jokaisessa opetuksessa korjauskerroin oli 0,1 ja pienin gradientti oli 0,0002. Monen aineiston opetuksessa ei koskaan saavutettu pienintä keskineliövirheen neliöjuurta, kun new-thyroid -aineistolla se saavutettiin usein, vaikka se oli asetettu pienemmäksi kuin muilla aineistoilla.

Taulukko 11. New-thyroid -aineiston opetusparametrejä.

	Epookit	Virhe
Opetus 1	1000	0,025
Opetus 2	1000	0,01
Opetus 3	2000	0,0001

Neuroverkko ei helposti ylioppinut new-thyroid -aineistoa. Esimerkiksi kun neuroverkolle opetti Buba-aineistoa, niin neuroverkko alkoi ylioppia, kun yritti pienentää opetuksen keskineliövirheen neliöjuurta opettamalla neuroverkkoa suuremmalla epookkien määrällä. New-thyroid -aineiston tapauksessa opetuksen keskineliövirheen neliöjuuren pieneneminen vähensi myös testiaineiston antamaa keskineliövirheen neliöjuurta selkeästi. Myös piiloneuroneita saatiin lisätä enemmän kuin esim. Buba-aineistossa ennen kuin neuroverkko alkoi ylioppia. Piiloneuroneja sai lisätä enemmän kuin nyrkkisäännön antama kolme piiloneuronia ja silti neuroverkon testiaineistoista antamien keskineliövirheiden neliöjuurien keskiarvo laski. Ero voidaan huomata tarkasteltaessa neuroverkkojen testiaineistoista antamien keskineliövirheiden neliöjuurien keskiarvoja eri piiloneuronien määrillä (kuva 31). Opetuksen 1 antama keskineliövirheiden neliöjuurien keskiarvo on lähes koko ajan suurempi kuin opetusten 2 ja 3 antamat arvot, vaikka opetuksessa 1 tavoiteltu virhe oli suurempi. Yleensä opetuksen virhettä ei tulisi laskea liikaa tai neuroverkko alkaa ylioppia aineistoa. Kuvasta 31 voidaan päätellä new-thyroid -aineistossa luokkien välillä olevan selkeät erot, jotka neuroverkko oppii sitä paremmin mitä enemmän sitä opetetaan.



Kuva 31. Opetusten 1 (o), 2 (x) ja 3 (+) testiaineistoista antamien keskineliövirheiden neliöjuurien keskiarvot eri piiloneuronien määrillä.

Kuvan 31 perusteella neuroverkon parhaassa arkkitehtuurissa on viisi piiloneuronia. Viiden piiloneuronin kohdalla jokaisessa kolmessa opetuksessa neuroverkkojen testiaineistoista antamien keskineliövirheiden neliöjuurien keskiarvo saavuttaa minimin. Lisäksi tarkastelemalla taulukoita 12, 13 ja 14 voidaan nähdä myös viiden piiloneuronin neuroverkon antaman tarkkuuden keskiarvon ja mediaanin olevan suurin.

Vaikka opetuksen 1 testiaineistosta antamat keskineliövirheiden neliöjuurien keskiarvot ovat korkeammat kuin opetuksilla 2 ja 3, on sen antama tarkkuus hyvä (taulukko 12). Neuroverkko pääsee parhaimmillaan 90 prosentin tarkkuuteen. Luokan *hyper* luokittelu on vaikeampaa opetukselle 1, kuin luokkien *normaali* ja *hypo* luokittelu.

Taulukko 12. New-thyroid -aineiston opetuksen 1 tuloksia kuudella eri arkkitehtuurilla (keskiarvo/mediaani).

	Tarkkuus	tpr_{hypo}	tpr_{hyper}	tpr_{normaali}
1 piiloneuroni	82%/81%	88%/100%	5%/0%	99%/100%
2 piiloneuronia	86%/86%	88%/100%	33%/25%	98%/100%
3 piiloneuronia	88%/90%	74%/100%	45%/50%	99%/100%
4 piiloneuronia	88%/86%	85%/100%	43%/46%	98%/100%
5 piiloneuronia	90%/90%	85%/100%	50%/50%	99%/100%
6 piiloneuronia	89%/90%	77%/100%	56%/50%	98%/100%

Opetuksessa 2 neuroverkkoa opetetaan enemmän kuin opetuksessa 1, sillä pienintä virhettä on pienennetty. Neuroverkon testiaineistoista antamien tarkkuuksien keskiarvo on parempi opetuksessa 2 kuin opetuksessa 1 (taulukko 13). Erot näkyvät yli yhden piiloneuronin arkkitehtuureissa. Enemmällä opettamisella neuroverkko oppii paremmin luokkia *hyper* ja *hypo*, joiden tpr -arvot ovat paremmat verrattuna opetukseen 1. Vertailemalla taulukkoja 12 ja 13 ja tarkastelemalla kuvaa 31 huomataan, että opetuksilla 2 ja 3 ei ole suurta eroa.

Taulukko 13. New-thyroid -aineiston opetuksen 2 tuloksia kuudella eri arkkitehtuurilla (keskiarvo/mediaani).

	Tarkkuus	tpr_{hypo}	tpr_{hyper}	tpr_{normaali}
1 piiloneuroni	81%/81%	84%/100%	10%/0%	98%/100%
2 piiloneuronia	93%/95%	94%/100%	76%/100%	98%/100%
3 piiloneuronia	93%/95%	92%/100%	84%/100%	97%/100%
4 piiloneuronia	96%/95%	89%/100%	94%/100%	98%/100%
5 piiloneuronia	96%/95%	92%/100%	88%/100%	98%/100%
6 piiloneuronia	95%/95%	92%/100%	87%/100%	98%/100%

Taulukko 14. New-thyroid -aineiston opetuksen 3 tuloksia kuudella eri arkkitehtuurilla (keskiarvo/mediaani).

	Tarkkuus	tpr_{hypo}	tpr_{hyper}	tpr_{normaali}
1 piiloneuroni	81%/81%	86%/100%	6%/0%	99%/100%
2 piiloneuronia	93%/95%	90%/100%	74%/100%	98%/100%
3 piiloneuronia	96%/95%	91%/100%	94%/100%	97%/100%
4 piiloneuronia	95%/95%	94%/100%	84%/100%	97%/100%
5 piiloneuronia	95%/95%	92%/100%	83%/100%	97%/100%
6 piiloneuronia	95%/95%	95%/100%	86%/100%	97%/100%

Vaikka aikaisemmat opetukset antoivat erittäin hyviä tuloksia, on syytä myös kokeilla opettaa neuroverkoille aineistoa vähän erilaisessa muodossa. Aikaisemmin käytetty tulosneuronin arvovälin [0, 1] jakaminen kolmeen osaan ei välttämättä ole tasa-arvoinen kaikille kolmelle luokalle. Lisäksi jos luokilla ei ole järjestystä, niin tämä ratkaisu ei tuota parasta mahdollista ratkaisua. Tätä varten kokeiltiin miten hyvin neuroverkko luokittelee new-thyroid -aineiston, jos luokka koodataan kuten *ecoli*- ja huimaus-aineistojen tapauksissa: kustakin luokasta tehdään oma tulosneuroni. Kolmen tulosneuronin käyttäminen rajoittaa nyrkkisäännön mukaan piiloneuronien määrän kahteen, sillä tällöin neuroverkon painokertoimien lukumäärä nousee.

Opetuksen tulokset kolmella tulosneuronilla opetuksella 2 ovat taulukossa 15. Neuroverkon antama tarkkuus ei muuttunut juuri yhtään kolmella tulosneuronilla. Erot yhden ja kolmen tulosneuronin opetuksissa voidaan huomata vertailemalla tpr_{hyper} -arvoja. Kolmen piiloneuronin tai sitä suuremman neuroverkon antamat tpr_{hyper} -arvot kolmella tulosneuronilla ovat aina yli 90 prosenttia (taulukko 15), kun yhdellä tulosneuronilla vastaavat luvut ovat 84-92 prosenttia (taulukko 13). Tästä voidaan päätellä, että luokan *hyper* luokittelu on heikompaa yhdellä tulosneuronilla. Tarkkuus ei ole muuttunut paljoa tpr_{hyper} -arvojen kasvusta huolimatta, koska luokan *normaali* suhteellinen frekvenssi on peräti 70 prosenttia ja siten luokka *normaali* vaikuttaa eniten tarkkuuteen.

Taulukko 15. New-thyroid -aineiston opetuksen 2 tuloksia viidellä eri arkkitehtuurilla kolmella tulosneuronilla (keskiarvo/ mediaani).

	Tarkkuus	tpr_{hypo}	tpr_{hyper}	tpr_{normaali}
2 piiloneuronia	93%/95%	79%/100%	84%/100%	98%/100%
3 piiloneuronia	95%/95%	88%/100%	91%/100%	98%/100%
4 piiloneuronia	96%/95%	91%/100%	94%/100%	98%/100%
5 piiloneuronia	97%/98%	92%/100%	92%/100%	98%/100%
6 piiloneuronia	95%/95%	90%/100%	92%/100%	97%/100%

Neuroverkko pystyi jopa 96 prosentin tarkkuudella luokittelemaan new-thyroid -aineistoa, mikä ylittää asetetun 70 prosentin rajan. Luokittelu onnistui mutkattomasti, vaikka new-thyroid -aineistossa oli vain 215 tapausta ja luokkien frekvenssit eivät olleet jakautuneet tasaisesti. Taulukoista 12, 13 ja 14 huomaa, että new-thyroid -aineiston luokitteluun tarvitaan monimutkaisempaa neuroverkkoa kuin kahden piiloneuronin neuroverkko. New-thyroid -aineiston luokitteluun tarvitaan siis vähintään kolme suoraa luokitteluun sitä. Selvän eron huomaa tarkastelemalla tpr_{hyper} -arvoja. Yhdellä piiloneuronilla tpr_{hyper} on korkeintaan 10 prosenttia, mutta piiloneuronien lisääminen parantaa varsinkin tpr_{hyper} -arvoja, mutta myös tpr_{hypo} -arvoja. Myös tpr_{normaali} on koko ajan korkea, koska 70 prosenttia tapauksista kuuluu luokkaan *normaali* ja siten neuroverkko oppii sen parhaiten. On myös selvää ettei yhdellä suoralla, eli yhdellä piiloneuronilla, pystytä jakamaan kolmea eri luokkaa. Kolmen luokan erottamiseen tasosta vaaditaan vähintään kaksi suoraa, eli neuroverkoilla kaksi piiloneuronia. Laittamalla neuroverkkoon jokaiselle kolmelle luokalle oma tulosneuronin saatiin vähän parannettua pienempien luokkien tpr -arvoja. Tämän tuloksen perusteella kolme tulosneuronin antaa paremman tuloksen new-thyroid -aineistosta kuin yksi tulosneuronin.

4.7. Pima

Pima-aineisto on suurin tässä tutkielmassa käytetyistä UCI-aineistoista [Blake and Merz, 1998] attribuuttien ja tapausten lukumäärien suhteen. Nämä kaksi ominaisuutta antavat hyvän mahdollisuuden neuroverkolle luokitteluun. Pima-aineistossa on kaksi luokkaa: diabetes (luokka *diabetes*) ja ei diabetes (luokka *normaali*). Luokkien frekvenssit ovat 500 (*normaali*) ja 268 (*diabetes*), joten tässäkin aineistossa frekvenssit eivät ole jakautuneet tasan. Pima-

aineiston tapauksessa neuroverkon antaman tarkkuuden tulisi olla yli 65 prosenttia tai saatu arvo on sama kuin mitä saadaan jos arvataan jokainen tapaus suurimpaan luokkaan.

Pima-aineistossa on kahdeksan attribuuttia, joten neuroverkkoon tulee kahdeksan syötoneuronia. Kaksi luokkaa pystytään koodaamaan yhdellä tulosneuronilla, jossa toinen luokka on 1 ja toinen 0. Koska jokaisessa opetusaineistossa tulee olemaan 691 tapausta koko aineiston 768 tapauksesta, niin nyrkkisäännön mukaan piiloneuroneja voisi olla korkeintaan kahdeksan.

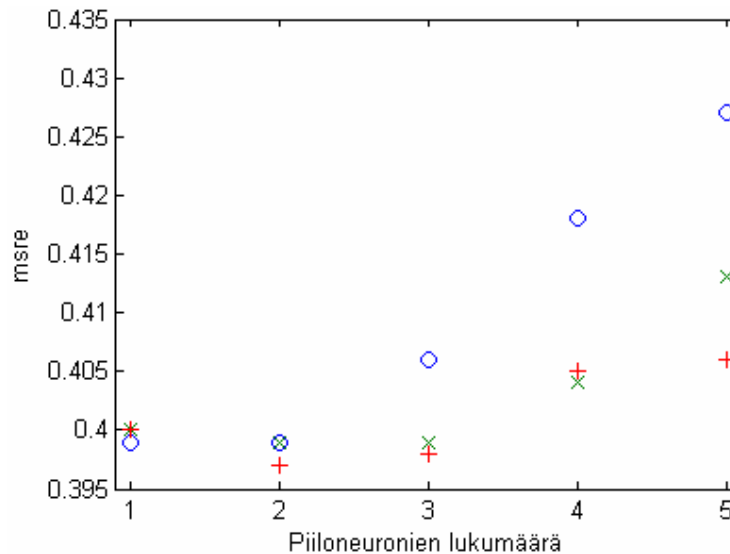
Neuroverkkujen opetukset tehtiin samalla tavalla kuin aiemmissakin aineistoissa. Opettaminen aloitettiin yhdellä piiloneuronilla ja niitä lisättiin yksi kerrallaan kunnes neuroverkon antama tarkkuus laski merkittävästi. Opetusparametrit olivat 10000 epookkia, 0,025 keskineliövirheen neliöjuuri, 0,1 korjauskerroin ja 0,0002 pienin gradientti.

Taulukko 16. Pima-aineiston antamat tarkkuudet ja *tpr*-arvot eri piiloneuronien määrillä testiaineistolle 1000 epookilla (keskiarvo/mediaani).

	Tarkkuus	<i>tpr</i> _{diabetes}	<i>tpr</i> _{normaali}
1 piiloneuroni	76%/78%	58%/60%	86%/87%
2 piiloneuronia	76%/76%	59%/59%	86%/86%
3 piiloneuronia	76%/76%	59%/60%	85%/84%
4 piiloneuronia	75%/74%	57%/58%	83%/83%
5 piiloneuronia	75%/75%	59%/60%	83%/83%

Pienemmällä piiloneuronien määrällä neuroverkon opettaminen loppui ennen 1000 epookkia, kun opetus saavutti pienimmän gradientin. Suuremmilla piiloneuronien määrällä opetus meni huomattavasti pidemmälle. Kun opetusparametriksi laittoi 1000 epookkia, neuroverkon tulos oli parempi kuin 10000 epookilla. 1000 epookilla neuroverkon laskemat tarkkuudet ja *tpr*-arvot olivat lähes samoja kuin 10000 epookilla (taulukko 16). Myös 500 epookin opetuksessa tarkkuudet olivat lähes samat. Tulosten erot voidaan huomata kuvan 32 kuvaajasta, josta näkee neuroverkon testiaineistosta antamien keskineliövirheiden neliöjuurien keskiarvojen erot 500, 1000 ja 10000 epookin opetuksissa. Erot näkyvät selkeästi yli kahden piiloneuronin neuroverkoissa, joissa opetus on jatkunut yli 1000 epookkia 10000 epookin opetuksessa. Vaikka

neuroverkkojen tarkkuudet ovat lähes samat, on niiden keskineliövirheiden neliöjuurissa eroja. 10000 epookin opetuksessa neuroverkko on alkanut ylioppia opetusaineistoa. Myös vertailtaessa 500 ja 1000 epookin opetuksia huomataan keskineliövirheiden neliöjuurien kasvun olevan 1000 epookin opetuksessa nopeampi kuin 500 epookin opetuksessa.



Kuva 32. 500 (+), 1000 (x) ja 10000 (o) epookin opetusten testiaineistosta antamien keskineliövirheiden neliöjuurien keskiarvot eri piiloneuronien määrillä. Jokaisella epookkien määrällä keskineliövirheen neliöjuuri saavuttaa minimin kahdella piiloneuronilla.

Taulukon 16 ja kuvan 32 perusteella voidaan päätellä kahden piiloneuronin neuroverkon olevan paras Pima-aineiston luokittelussa, sillä sen jälkeen keskineliövirheen neliöjuuri ei laske missään opetuksessa. 76 prosentin tarkkuus on hyvä tulos. Hyvään tulokseen auttoivat ilmeisesti aineiston suuruus ja attribuuttien määrä.

4.8. Virtsainkontinenssi

Virtsainkontinenssi-aineistossa on 529 tapausta. Tässä tutkielmassa käytetyssä aineistossa on kolme luokkaa: *stress*, *mixed* ja *muut*. Luokkien frekvenssit ovat 323 (*stress*), 140 (*mixed*) ja 66 (*muut*), joten luokkien jakauma on vino myös tässä aineistossa. Virtsainkontinenssi-aineiston tapauksessa neuroverkon antaman tarkkuuden tulisi olla yli 61 prosenttia, jotta neuroverkon antama tulos olisi parempi kuin mitä saadaan, jos arvataan jokainen tapaus suurimpaan luokkaan.

Virtsainkontinenssi-aineistoa on aikaisemmin myös opetettu neuroverkoille. Neuroverkossa oli neljä piiloneuronia ja kolme tulosneuronia.

Opetuksessa käytettiin kymmenkertaista ristiinvalidointia. Näiden neuroverkkojen antamat tulokset ovat taulukossa 17. [Laurikkala, 2001a]

Taulukko 17. Aikaisempia tuloksia virtsainkontinenssi-aineiston opettamisesta neuroverkoille. [Laurikkala, 2001a]

Tarkkuus	78%
tpr_{stress}	88%
tpr_{mixed}	65%
tpr_{muut}	51%

Virtsainkontinenssi-aineistossa on seitsemän attribuuttia, joten neuroverkkoon tulee seitsemän syötoneuronia. Tulosneuronien suhteen virtsainkontinenssi-aineistolla on tehty samalla tavalla kuin new-thyroid -aineiston kanssa. Aluksi on käytetty yhtä tulosneuronia ja sen jälkeen kolmea tulosneuronia. Yhden tulosneuronin tapauksessa nyrkkisäännön mukaan neuroverkossa voisi olla korkeintaan kuusi piiloneuronia, jotta neuroverkko voisi oppia aineiston. Kolmen tulosneuronin tapauksessa piiloneuronien lukumäärä voisi nyrkkisäännön mukaan olla korkeintaan viisi.

Yhden tulosneuronin neuroverkossa luokkien arvoksi tulee 0, 0,5 ja 1. Tulosneuronin arvoväli [0.0, 1.0] on jaettu kolmeen osaan siten, että jokaisella luokalla on yhtä suuri alue tulosneuronin arvoväliltä: [0.0, 0.33], (0.33, 0.67) ja [0.67, 1.0]. Tämä tulosneuronin jako on sama kuin mitä käytettiin new-thyroid -aineiston kanssa.

Aluksi opetettiin neuroverkkoja, joissa oli yksi tulosneuron. Eri opetuksien opetusparametrit ovat taulukossa 18. Jokaisessa opetuksessa korjauskerroin oli 0,1 ja pienin gradientti 0,0002. Opetukset aloitettiin kahden piiloneuronin neuroverkolla, mikä on pienin piiloneuroneiden lukumäärä kolmen luokan luokitteluun. Tämän jälkeen lisättiin piiloneuronien lukumäärää, kunnes neuroverkon testiaineistoille antaman keskineliövirheen neliöjuuren keskiarvo alkoi kasvaa tai ei enää laskenut paljoa. Jo aikaisemmista virtsainkontinenssi-aineistosta tehtyjen neuroverkkojen ([Laurikkala, 2001a]) perusteella oli odotettavissa, että neuroverkko pystyisi oppimaan virtsainkontinenssi-aineiston hyvin.

Taulukko 18. Virtsainkontinenssi-aineiston opetusparametrejä.

	Epookit	Virhe
Opetus 1	500	0,025
Opetus 2	500	0,01
Opetus 3	1000	0,01

Taulukossa 19 on opetuksen 1 antamat tulokset yhden tulosneuronin neuroverkoilla. Sen perusteella voidaan heti huomata, että tulokset ovat parempia kuin mitä Laurikkala [2001a] oli saanut samasta aineistosta neuroverkoilla (taulukko 17). Jälleen voidaan huomata, että suurimman luokan (*stress*) *tpr*-arvo on selvästi paras jokaisella neuroverkon arkkitehtuurilla. Opetuksella 1 tarkkuudet ja *tpr*-arvot eivät paljoa muuttuneet eri piiloneuronien lukumäärillä, varsinkaan luokilla *stress* ja *mixed*. Kuten muillakin tässä tutkielmassa käytetyillä aineistoilla, neuroverkko oppi huonoiten pienimmän luokan (*muut*) virtsainkontinenssi-aineistossakin.

Taulukko 19. Virtsainkontinenssi-aineiston opetuksen 1 tuloksia kuudella eri arkkitehtuurilla yhdellä tulosneuronilla (keskiarvo/mediaani).

	Tarkkuus	<i>tpr</i> _{muut}	<i>tpr</i> _{mixed}	<i>tpr</i> _{stress}
2 piiloneuronia	87%/87%	76%/86%	78%/81%	93%/94%
3 piiloneuronia	87%/88%	75%/78%	78%/78%	94%/95%
4 piiloneuronia	88%/88%	72%/78%	79%/79%	93%/94%
5 piiloneuronia	88%/88%	76%/83%	79%/80%	94%/94%
6 piiloneuronia	88%/88%	75%/75%	78%/81%	94%/94%
7 piiloneuronia	87%/87%	69%/75%	78%/78%	94%/95%

Opetuksella 1 neuroverkon opettaminen päättyi usein aikaisin opetuksen saavuttaessa tavoitellun virheen 0,025. Tämän perusteella neuroverkko voisi pystyä oppimaan vielä paremmin kuin opetuksessa 1. Opetuksessa 2 tavoiteltu virhe on pienempi kuin opetuksessa 1 ja täten neuroverkkoa on opetettu

enemmän. Opetuksen 2 antamat tulokset ovat taulukossa 20. Vertailemalla opetuksia 1 ja 2 ei paljoa eroja löydy. Tarkkuudet ovat molemmissa noin 88 prosenttia. Ainoa ero opetuksissa 1 ja 2 on tpr_{muut} -arvoissa. Opetuksessa 2 tpr_{muut} on parempi suuremmilla piiloneuronien määrillä kuin opetuksessa 1.

Taulukko 20. Virtsainkontinenssi-aineiston opetuksen 2 tuloksia kuudella eri arkkitehtuurilla yhdellä tulosneuronilla (keskiarvo/mediaani).

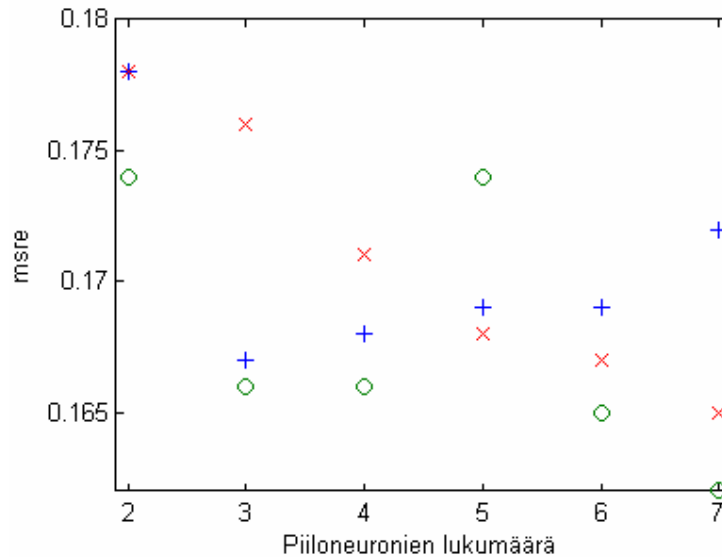
	Tarkkuus	tpr_{muut}	tpr_{mixed}	tpr_{stress}
2 piiloneuronia	88%/88%	73%/80%	79%/80%	94%/94%
3 piiloneuronia	88%/88%	71%/73%	79%/77%	95%/96%
4 piiloneuronia	88%/87%	73%/71%	79%/76%	94%/94%
5 piiloneuronia	87%/88%	71%/78%	78%/81%	94%/94%
6 piiloneuronia	88%/88%	80%/80%	78%/80%	95%/94%
7 piiloneuronia	88%/88%	82%/80%	77%/79%	95%/97%

Opetuksella 3 neuroverkkoa opetettiin vielä enemmän kuin opetuksella 2, sillä opetuksessa 3 epookkien määrä on suurempi. Opetuksessa 3 neuroverkon antamat tarkkuudet (taulukko 21) olivat lähes samoja kuin opetuksilla 1 ja 2. Opetuksilla 2 ja 3 oli samantyylinen ero kuin opetuksilla 1 ja 2 oli: tpr_{muut} on opetuksella 3 parempi kuin opetuksella 2. Yleinen suuntaus virtsainkontinenssi-aineistolla näyttäisi olevan, että mitä enemmän neuroverkkoa opetetaan ja mitä enemmän siinä on piiloneuroneja, sitä paremmin se pystyy luokittelemaan luokkaa *muut*.

Taulukko 21. Virtsainkontinenssi-aineiston opetuksen 3 tuloksia kuudella eri arkkitehtuurilla yhdellä tulosneuronilla (keskiarvo/mediaani).

	Tarkkuus	tpr_{muut}	tpr_{mixed}	tpr_{stress}
2 piiloneuronia	87%/88%	68%/74%	82%/82%	94%/94%
3 piiloneuronia	87%/87%	74%/71%	78%/80%	94%/96%
4 piiloneuronia	88%/88%	76%/75%	77%/77%	94%/94%
5 piiloneuronia	88%/88%	80%/80%	79%/78%	95%/95%
6 piiloneuronia	88%/88%	79%/75%	78%/80%	94%/94%
7 piiloneuronia	88%/88%	83%/87%	78%/81%	94%/94%

Koska jokaisella kolmella opetuksella tarkkuudet ja tpr -arvot ovat lähellä toisiaan, niin on vaikea tehdä päätöstä, mikä opetus ja piiloneuronien lukumäärä olisi paras mahdollinen virtsainkontinenssi-aineiston luokitteluun yhdellä tulosneuronilla. Tarkastelemalla neuroverkon testiaineistoista antamien keskineliövirheiden neliöjuurien keskiarvoja voidaan huomata, milloin neuroverkko alkaa mahdollisesti ylioppia ja mikä olisi näin paras opetus ja arkkitehtuuri. Tarkastelemalla kuvaa 33 voidaan huomata, että jokaisella opetuksella piiloneuronien lukumäärä vaikuttaa keskineliövirheiden neliöjuurien keskiarvoon eritavalla. Opetuksella 1 keskineliövirheiden neliöjuurien keskiarvo kasvaa kolmen piiloneuronin jälkeen, kun taas opetuksella 3 se laskee koko ajan. Opetuksilla 1 ja 2 keskineliövirheiden neliöjuurien keskiarvo kasvaa tai ei laske paljoa kolmen piiloneuronin jälkeen, joten näillä opetuksilla optimaalinen piiloneuronien määrä on kolme. Opetuksen 3 optimaalinen piiloneuronien lukumäärä olisi kuvan 33 mukaan seitsemän. On syytä katsoa myös taulukon 21 tuloksia ja yhdistää molemmat tulokset. Tällä tavalla optimaaliseksi piiloneuronien lukumääräksi opetuksella 3 saadaan myös seitsemän, sillä seitsemällä piiloneuronilla neuroverkon testiaineistoista antama keskineliövirheiden neliöjuurien keskiarvo on pienin sekä tarkkuus ja tpr -arvot ovat parhaat. Lisäksi seitsemän piiloneuronia on lähellä nyrkkisäännön mukaista suurinta piiloneuronien lukumäärää, joka on kuusi.



Kuva 33. Opetusten 1(+), 2 (o) ja 3 (x) testiaineistoista antamien keskineliövirheiden neliöjuurien keskiarvot eri piiloneuronien lukumäärillä yhden tulosneuronin neuroverkosta. Opetuksen 2 viiden piiloneuronin tulosta voidaan pitää poikkeavana arvona.

Seuraavaksi opetettiin neuroverkkoja, joissa oli kolme tulosneuria. Kukin tulosneuri kertoo, kuinka paljon tapaus muistuttaa kyseistä luokkaa. Neuroverkon katsotaan luokittelevan tapauksen siihen luokkaan, jonka tulosneuronin antama tulos on suurin. Kolmella tulosneuronilla ei tule ongelmia luokkien järjestyksen kanssa, kuten yhden tulosneuronin kanssa.

Kolmen tulosneuronin neuroverkkojen opettaminen aloitettiin 200 epookilla ja 0,04 virheellä muiden opetusparametrien ollessa samoja kuin aikaisemmin. Kolmella tulosneuronilla keskineliövirheen neliöjuuri ei mennyt niin pieneksi kuin se meni yhden tulosneuronin neuroverkoissa. Keskineliövirheen neliöjuuri meni opetuksessa vähän alle 0,04 kun yhdellä tulosneuronilla sama luku oli lähes 0,02. 200 epookin opetuksella tarkkuus parani keskimäärin prosenttiyksikön. Tällöin tpr_{mixed} -arvot olivat keskimäärin korkeampia ja tpr_{muut} -arvot matalampia kuin yhden tulosneuronin neuroverkossa. Tämän perusteella kolmen tulosneuronin neuroverkko on parempi luokittelemaan luokkaa *stress* ja huonompi luokkaa *muut* kuin yhden tulosneuronin neuroverkko. Neuroverkon antamat tulokset ovat taulukossa 22.

Taulukko 22. Virtsainkontinenssi-aineiston tuloksia kuudella eri arkkitehtuurilla kolmella tulosneuronilla 200 epookin opetuksella (keskiarvo/mediaani).

	Tarkkuus	tpr_{muut}	tpr_{mixed}	tpr_{stress}
2 piiloneuronia	86%/87%	58%/63%	83%/85%	93%/94%
3 piiloneuronia	89%/88%	66%/71%	85%/87%	93%/93%
4 piiloneuronia	88%/90%	71%/67%	85%/85%	94%/96%
5 piiloneuronia	88%/90%	74%/71%	84%/86%	94%/97%
6 piiloneuronia	89%/88%	73%/75%	85%/85%	94%/95%
7 piiloneuronia	89%/90%	76%/78%	85%/83%	94%/94%

Seuraavaksi kokeiltiin opettaa neuroverkkoa lisää ja asetettiin epookkien lukumääräksi 500 muiden opetusparametrien ollessa samoja kuin edellisessä opetuksessa. Nyt varsinkin suuremmilla piiloneuronien lukumäärillä neuroverkko saavutti opetuksen aikana tavoitellun virheen 0,04. Neuroverkon antama tarkkuus oli 87 ja 90 prosentin välillä, mikä oli prosenttiyksikön parempi kuin 200 epookin opetuksessa (taulukko 22). Muuten tpr -arvot olivat lähes samoja. Eroa ei ollut siis kovin paljoa 200 ja 500 epookin opetuksissa.

Neuroverkkoa opetettiin vielä vähän enemmän asettamalla tavoitelluksi virheeksi 0,025 ja muut opetusparametrit pidettiin samoina kuin edellisessä opetuksessa. Opetuksen tulokset olivat lähes samat kuin edellisessä opetuksessa, jossa tavoiteltu virhe oli vähän suurempi 0,04.

Kaikilla kolmella opetuskerralla kolmen tulosneuronin neuroverkoilla tarkkuudet ja tpr -arvot olivat lähes samoja. Kolmen tulosneuronin neuroverkoissa tpr_{mixed} -arvot olivat parempia ja tpr_{muut} -arvot huonompia kuin yhden tulosneuronin neuroverkoissa. Molemmissa nousu ja lasku olivat lähes samansuuruisia, joten ei voida sanoa suoraan, onko yhden vai kolmen tulosneuronin neuroverkko parempi.

Neuroverkko oppi keskimäärin virtsainkontinenssi-aineiston 88 prosentin tarkkuudella. Tämä ylittää selvästi asetetun 61 prosentin rajan. Tässä saatu tarkkuus oli suurin piirtein sama mitä Laurikkala [2001b] oli saanut k_n lähimmän naapurin menetelmällä, n. 90 prosenttia. Lisäksi se on jopa 10 prosenttiyksikköä korkeampi kuin Laurikkalan [2001a] saamat tulokset samasta aineistosta neuroverkoilla. Virtsainkontinenssi-aineistossa, kuten

muissakin tässä tutkielmassa käytetyistä aineistoissa, neuroverkko oppi parhaiten suurimman luokan. Riippuu käyttökohteesta onko parempi käyttää yhden vai kolmen tulosneuronin neuroverkkoa. Tulosneuronien lukumäärä tulisi valita kiinnostuksena olevan kohteen mukaan. Mikäli kiinnostuksen kohteena on enemmän luokka *muut* kuin luokka *mixed*, niin olisi parempi valita yhden tulosneuronin neuroverkko, sillä tällä neuroverkolla tpr_{mixed} oli parempi kuin kolmen tulosneuronin neuroverkolla. Vastaavasti jos kiinnostuksen kohteena on enemmän luokka *mixed* kuin luokka *muut*, niin olisi parempi valita kolmen tulosneuronin neuroverkko, sillä tällä neuroverkolla tpr_{mixed} oli parempi kuin yhden tulosneuronin neuroverkolla.

5. Tapausten keinotekoinen lisääminen

Neuroverkko tulisi opettaa tarpeeksi suurella aineistolla, jotta se pystyisi oppimaan aineiston piirteet. Toisaalta myös vino luokkajakauma saattaa haitata neuroverkon oppimista, kuten huomattiin luvussa 4. Koska alokas-aineistoa lukuun ottamatta kaikki tässä tutkielmassa käytetyt aineistot ovat olleet liian pieniä neuroverkoille, on käytetty ristiininvalidointia. Mikäli aineisto on kovin pieni, niin voidaan tehdä ristiininvalidointi niin monta kertaa kuin aineistossa on tapauksia (leave-one-out) [Swingler, 1996]. Tällöin vain yksi tapaus muodostaisi aina testiaineiston ja loput aineistosta olisi opetusaineistona. Näin testiaineisto olisi kuitenkin niin pieni, ettei tätä tapaa kannata käyttää neuroverkoilla.

Swingler [1996] ehdottaa pienten aineistojen tapauksissa, että aineistoon voitaisiin lisätä keinotekoisesti tapauksia ja käyttää näin saatuja keinotekoisesti suurennettuja aineistoja. Luokittelussa luokkien vinoa jakaumaa pystytään tasaamaan lisäämällä tapauksia pienempiin luokkiin. Näin on myös mahdollista ulottaa syöteavaruus paikkoihin, joihin ei tavallisesti päästä [Swingler, 1996]. Tällä tarkoitetaan sitä, että saadaan tapauksia, joita ei aineistossa jo ennestään ole. Kun attribuuttien arvot arvotaan keinotekoisesti, niin voidaan saada esim. kolmen attribuutin yhdistelmä, jonka kaltaista ei ole alkuperäisessä aineistossa.

Tässä luvussa on lisätty keinotekoisesti tapauksia huimaus-, Pima- ja virtsainkontinenssi-aineistoihin ja opetettu näitä keinotekoisesti suurennettuja aineistoja neuroverkoille. Kahdeksasta aineistosta vain kolmeen aineistoon lisättiin tapauksia, koska sen katsottiin olevan sopiva määrä aineistojen keinotekoisien suurentamisen vaikutuksen tutkimiseksi tämän tutkielman puitteissa. Huimaus- ja virtsainkontinenssi-aineisto valittiin keinotekoisesti suurennettaviksi, koska niitä on tutkittu Suomessa jo aiemmin [Laurikkala 2001a; Viikki 2002] ja molemmissa aineistoissa on vaikeasti ennustettavia pieniä luokkia. Pima-aineisto valittiin kolmanneksi aineistoksi, koska sen suuruus antaa hyvät lähtökohdat neuroverkoille oppimiseen. Aineistoissa on lisätty kaikkiin paitsi suurimpaan luokkaan uusia tapauksia keinotekoisesti. Tällä tavalla aineistojen luokkajakauman vinoutta on saatu vähennettyä. Aineistoihin on lisätty 20 ja 30 prosenttia uusia tapauksia kuhunkin luokkaan ja testattu miten se vaikuttaa neuroverkkojen oppimiseen. 20 prosentin lisäyksen jälkeen noin 17 prosenttia luokan tapauksista on keinotekoisesti lisättyjä. Vastaava arvo 30 prosentin lisäyksen jälkeen on noin 23 prosenttia. Tapauksia ei kannata lisätä keinotekoisesti liikaa, koska tällöin aineisto ei olisi enää luonnollinen ja vääristäisi aineistoa (vrt. puuttuvien arvojen korvaaminen). Tapauksia on

lisätty aina kyseisen luokan jakauman mukaan MATLAB-ohjelmiston `normrnd`-funktiolla. `Normrnd`-funktio antaa satunnaisluvun attribuutin jakaumasta parametreinä saamiensa keskiarvon ja keskihajonnan perusteella. Tällä tavalla jokainen keinotekoisesti lisätty tapaus noudattaa karkeasti luokkansa jakaumaa. Tarkempia jakauma-approksimaatioita olisi saatu, mikäli kunkin attribuutin jakauma olisi tunnistettu ja keinotekoisia arvoja olisi tuotettu näiden jakaumien tunnuslukujen avulla. Tässä tutkielmassa käytettiin normaalijakaumaa, koska sen arveltiin olevan lähellä attribuuttien jakaumia.

Jakauman vinoutta voidaan mitata erilaisilla menetelmillä. Tässä luvussa jokaisen aineiston kohdalla on laskettu luokkamuuttujan jakauman vinous MATLAB-ohjelmiston tilastopakettin `skewness`-funktiolla. Mitä pienempi funktion antaman arvon itseisarvo on, sitä vähemmän jakauma on vino. Vastaavasti mitä suurempi itseisarvo on, sitä enemmän jakauma on vino. [MATLAB, 2002]

Koska luvussa 4 on samoille aineistoille löydetty parhaat opetusparametrit ja piiloneuronien lukumäärät, ei niitä tarvitse enää etsiä. Neuroverkkoja on opetettu käyttämällä keinotekoisesti suurennettuja aineistoja luvussa 4 löydettyillä optimaalisilla opetusparametreilla ja piiloneuronien lukumäärillä, ja on tutkittu miten tapausten keinotekoinen lisääminen vaikuttaa neuroverkkojen oppimiseen.

Luvussa 4 neuroverkkoja opettaessa tehtiin yhden kerran kymmenkertainen ristiinvalidointi, jossa jokaisella ristiinvalidointikerralla neuroverkkoa opetettiin kymmenen kertaa. Tällöin jokaista neuroverkkoa opetettiin yhteensä 100 kertaa ja otettiin muistiin näiden tulosten keskiarvo ja mediaani. Tuloksista saadaan realistisemmat, jos tehdään kymmenkertainen ristiinvalidointi kymmenen kertaa. Näin saatavat tulokset ovat luotettavampia, koska opetuskertoja on enemmän erilaisilla opetus- ja testijoukoilla, ja kahden eri opetuksen eroja voidaan verrata keskenään, kun tuloksista otetaan keskiarvo ja mediaani. Siksi tässä luvussa neuroverkkoja opettaessa on tehty kymmenkertainen ristiinvalidointi kymmenen kertaa, jossa jokaisella ristiinvalidointikerralla neuroverkkoja on opetettu kymmenen kertaa. Tällöin jokaista neuroverkkoa opetetaan yhteensä 1000 kertaa. Myös alkuperäiset aineistot on opetettu tässä luvussa uudestaan neuroverkoille käyttäen kymmenkertaista ristiinvalidointia kymmenen kertaa, jotta alkuperäisten ja keinotekoisesti suurennettujen aineistojen opetukset olisivat vertailukelpoisia.

Alkuperäisillä ja keinotekoisesti suurennetuilla aineistoilla saatuja tuloksia on verrattu toisiinsa tarkkuuksien ja *tpr*-arvojen keskiarvojen vertailemisen lisäksi *Mann-Whitneyn U-testillä* (Mann-Whitney's U-test) (jatkossa U-testi). U-testin hypoteesina on, että kaksi riippumatonta yhden muuttujan otosta on

peräisin samasta populaatiosta (H_0 : otos A ja otos B ovat peräisin samasta populaatiosta X, H_1 : otos A ja otos B ovat peräisin eri populaatioista). Kun tehdään U-testi alkuperäisen ja 20 prosentin lisäyksen aineistojen tarkkuuksille, niin testin tulos kertoo kuinka todennäköisesti näiden tarkkuudet ovat samat (millä todennäköisyydellä H_0 on tosi). Tällä tavalla pystytään selvittämään, onko alkuperäisen ja keinotekoisesti suurennettun aineiston tarkkuuksilla ja *tpr*-arvoilla tilastollisesti merkitsevää eroa. On huomattava, että tässä U-testin oletukset eivät täysin täyty. Kymmenen kertaa toistetussa kymmenkertaisessa ristiinvaldoinnissa testijoukot eivät enää ole täysin erillisiä ja tästä syystä ajon tulokset eivät ole toisistaan riippumattomia. Lisäksi otokset A ja B eivät ole täysin keskenään riippumattomia, sillä keinotekoisesti suurennetuissa aineistoissa on mukana edelleen myös alkuperäisen tapaukset. Tämän takia U-testien antamia tuloksia tulee tulkita varovasti. [Pett, 1997]

Mikäli verrataan aiempia luvun 4 tuloksia toisiinsa, voidaan tehdä hypoteesi miten tapausten keinotekoinen lisääminen aineistoihin vaikuttaa neuroverkon oppimiseen. Luvussa 4 neuroverkko oppi yleensä parhaiten suurimmat luokat ja pienimmät luokat se oppi huonommin. Tämän perusteella tapausten keinotekoinen lisääminen ja sitä kautta luokkien jakauman tasoittaminen parantaa pienempien luokkien luokittelua ja *tpr*-arvoja ja vastaavasti huonontaa suurimman luokan luokittelua ja *tpr*-arvoa.

5.1. Tapausten keinotekoinen lisääminen huimaus-aineistoon

Huimaus-aineiston luokkien absoluuttiset ja suhteelliset frekvenssit näkyvät taulukossa 23. Tapausten keinotekoisella lisäämisellä saadaan suurimman luokan (*Meniere*) keskimääräistä suhteellisen frekvenssin eroa muihin luokkiin vähennettyä seitsemän prosenttiyksikköä. Muiden kuin luokan *Meniere* suhteellinen frekvenssi ei muutu paljoa, mutta luokan *Meniere* suhteellisen frekvenssin vähentyminen tasoittaa luokkajakaumaa. Luokkamuuttujan vinous on alkuperäisellä aineistolla -0,52, 20 prosenttia lisätyllä aineistolla -0,49 ja 30 prosenttia lisätyllä aineistolla -0,47. Luokkamuuttujan vinous laskee koko ajan, mutta lasku on pientä. Luokka *Meniere* jää edelleen selvästi muita luokkia suuremmaksi, mutta frekvenssien erot ovat nyt pienempiä ja luokkajakauma tasaisempi kuin alkuperäisellä aineistolla. Alkuperäisen aineiston koko on 815 tapausta, 20 prosentin lisäyksellä aineiston koko on 915 tapausta ja 30 prosentin lisäyksellä aineiston koko on 966 tapausta. Huimaus-aineisto opetettiin neuroverkolle, jossa oli neljä piiloneuronia. Tämä piiloneuronien määrä katsottiin olevan optimaalinen huimaus-aineiston luokitteluun luvun 4.5 perusteella.

Taulukko 23. Huimaus-aineiston frekvenssit alkuperäiselle aineistolle sekä 20 ja 30 prosenttia pienemmän luokan tapauksia lisättyihin aineistoihin.

Aineisto	Absoluuttinen frekvenssi (suhteellinen)					
	<i>an</i>	<i>tv</i>	<i>sd</i>	<i>Meniere</i>	<i>bpv</i>	<i>vn</i>
Alku- peräinen	120 (15%)	65 (8%)	41 (5%)	313 (38%)	146 (18%)	130 (16%)
20% lisäys	144 (16%)	78 (9%)	49 (5%)	313 (34%)	175 (19%)	156 (17%)
30% lisäys	156 (16%)	85 (9%)	53 (5%)	313 (32%)	190 (20%)	169 (18%)

Alkuperäisen, 20 prosentin ja 30 prosentin lisäysten opetusten tarkkuudet ja *tpr*-arvot ovat taulukossa 24. Mitä enemmän aineistossa on tapauksia, sitä parempi on neuroverkkojen antama tarkkuus. Tarkkuuksien erot ovat kuitenkin hyvin pieniä, joten ei voida tehdä johtopäätöstä, että tarkkuus paranisi lisäämällä huimaus-aineistoon tapauksia keinotekoisesti.

Taulukko 24. Huimaus-aineiston antamat tarkkuudet ja *tpr*-arvot erikokoisilla aineistoilla (keskiarvo/mediaani).

Aineisto	Tarkkuus	tpr_{an}	tpr_{tv}	tpr_{sd}	$tpr_{Meniere}$	tpr_{bpv}	tpr_{vn}
Alku- peräinen	80%/80%	87%/89%	78%/82%	12%/0%	89%/90%	69%/69%	83%/85%
20% lisäys	80%/80%	89%/92%	82%/86%	13%/0%	87%/88%	72%/73%	86%/88%
30% lisäys	81%/81%	91%/93%	85%/88%	13%/0%	86%/87%	73%/74%	87%/89%

Tarkastellaan neuroverkkojen antamia tuloksia tarkemmin U-testillä. Testien antamat tulokset ovat taulukossa 25. Testien tulosten mukaan alkuperäisen aineiston antamat tarkkuudet ja *tpr*-arvot ovat täysin erilaiset kuin aineistoilla, joihin on lisätty tapauksia keinotekoisesti. Tämä osoittaa, ainakin

U-testin perusteella, että tapausten keinotekoinen lisääminen vaikuttaa huimaus-aineistolle sekä tarkkuuteen että tpr -arvoihin. 20 ja 30 prosentin lisäyksilläkin tarkkuuksien ja tpr -arvojen erot ovat enimmäkseen hyvin suuria. Ainoastaan tpr_{sd} on lähes sama molemmilla lisätyillä aineistoilla ja lisäksi tpr_{bvp} :t ovat jollakin tavalla samanlaisia. Koska U-testin perusteella tarkkuudet eivät ole samoja, voidaan päätellä taulukon 24 perusteella, että keinotekoisesti suurennetulla aineistolla on vähän parempi tarkkuus kuin alkuperäisellä aineistolla.

Taulukko 25. U-testien tulokset huimaus-aineistosta. Tulos kertoo kuinka suurella todennäköisyydellä kahdesta eri aineistosta saadut tulokset ovat samat.

Aineisto	Tarkkuus	tpr_{an}	tpr_{tv}	tpr_{sd}	$tpr_{Meniere}$	tpr_{bvp}	tpr_{vn}
Alkuperäinen - 20% lisäys	3%	0%	0%	3%	0%	0%	0%
Alkuperäinen - 30% lisäys	0%	0%	0%	1%	0%	0%	0%
20% lisäys - 30% lisäys	6%	0%	1%	59%	0%	27%	0%

Huimaus-aineiston kohdalla hypoteesi (tapausten keinotekoinen lisääminen nostaa pienempien luokkien tpr -arvoja ja laskee suurimman luokan tpr -arvoa) pitää paikkaansa lähes jokaisen luokan kohdalla. Vähiten on muuttunut tpr_{sd} , joka on noussut vain yhden prosenttiyksikön ja sen arvo on edelleen hyvin pieni. Arvo tpr_{sd} on edelleen hyvin pieni, joten neuroverkko ei pysty luokittelemaan sitä juuri lainkaan, vaikka ko. luokkaan lisättäisiin keinotekoisesti tapauksia. Kun muihin luokkiin on lisätty keinotekoisesti tapauksia, niin $tpr_{Meniere}$ on laskenut kolme prosenttiyksikköä. Se on kuitenkin edelleen hyvin korkea, 86 prosenttia, myös keinotekoisesti suurennetuilla aineistoilla. Ainoa luokka johon tapausten keinotekoinen lisääminen ei vaikuttanut, oli sd , sillä tpr_{sd} pysyi jokaisella kolmella aineistolla lähes samana. Muiden luokkien (an , tv , bvp ja vn) tpr -arvot nousivat 2-7 prosenttiyksikköä keinotekoisesti suurennetuilla aineistoilla verrattuna alkuperäiseen aineistoon.

Huimaus-aineistossa tapausten keinotekoinen lisääminen tuotti hypoteesin mukaisen tuloksen, eli pienempien luokkien luokittelu parani, kun niihin

lisättiin keinotekoisesti tapauksia. Mitä enemmän tapauksia lisättiin luokkaan, sitä paremmin ko. luokka saatiin luokiteltua. Poikkeuksena oli jo entuudestaan vaikeasti luokiteltava luokka *sd*, jonka *tpr*-arvojen keskiarvo ei muuttunut juuri yhtään. Samalla suurimman luokan (*Meniere*) luokittelu huononi, joten tarkkuuksien keskiarvo ei juuri muuttunut. Sen sijaan $tpr_{Meniere}$ huononi sitä enemmän mitä enemmän tapauksia lisättiin muihin luokkiin eli mitä pienempi luokan *Meniere* suhteellinen frekvenssi oli, sitä huonompi $tpr_{Meniere}$ oli. Neuroverkon testiaineistoista antamien keskineliövirheiden keskiarvo oli jokaisella aineistolla n. 0,23, joten neuroverkko oppi jokaisen aineiston tässä suhteessa yhtä hyvin. Koska sekä tarkkuuksien keskiarvo että keskineliövirheen neliöjuuri pysyivät lähes samoina jokaisella aineistolla, ei tapausten keinotekoisella lisäämisellä huimaus-aineistoon ole juuri vaikutusta tarkkuuksien keskiarvon ja keskineliövirheiden neliöjuuren suhteen. U-testin tulosten perusteella (taulukko 25) eri aineistojen väliset tarkkuudet eivätkä *tpr*-arvot ole peräisin samasta otoksesta, vaikka niiden keskiarvot ja mediaanit ovat lähes samoja.

Mikäli huimaus-aineiston kohdalla mielenkiinnon kohteena on jokin luokista *an*, *tv*, *bpv* tai *vn*, niin tapausten keinotekoinen lisääminen parantaa näiden luokkien luokittelua. Riippuu mielenkiinnon kohteesta, onko kannattavaa lisätä keinotekoisesti tapauksia huimaus-aineistoon.

5.2. Tapausten keinotekoinen lisääminen Pima-aineistoon

Pima-aineiston luokkien absoluuttiset ja suhteelliset frekvenssit, sekä *vinous* alkuperäiselle ja kahdelle keinotekoisesti suurennetulle aineistolle näkyvät taulukossa 26. Tapausten keinotekoisella lisäämisellä saadaan kahden luokan suhteellisen frekvenssin erotusta vähennettyä 30 prosenttiyksiköstä jopa 18 prosenttiyksikköön. Luokkamuuttujan *vinouden* lähestyminen nolaa kertoo myös jakauman selkeästä tasoittumisesta. Luokkamuuttujan jakauma jää silti edelleen melko *vinoksi*. Alkuperäisen aineiston koko on 768 tapausta, 20 prosentin lisäyksellä aineiston koko on 822 tapausta ja 30 prosentin lisäyksellä aineiston koko on 848 tapausta. Pima-aineisto opetettiin neuroverkolle, jossa oli kaksi piiloneuronia. Piiloneuronien määrä katsottiin olevan optimaalinen Pima-aineiston luokitteluun luvun 4.7 perusteella.

Taulukko 26. Pima-aineiston frekvenssit ja vinoudet alkuperäiselle aineistolle sekä 20 ja 30 prosenttia pienemmän luokan tapauksia lisättyihin aineistoihin.

Aineisto	Absoluuttinen frekvenssi (suhteellinen)		Vinous
	<i>Diabetes</i>	<i>Normaali</i>	
Alkuperäinen	500 (65%)	268 (35%)	0,63
20% lisäys	500 (61%)	322 (39%)	0,44
30% lisäys	500 (59%)	348 (41%)	0,36

Alkuperäisellä aineistolla saatu tarkkuus oli 77 prosenttia, $tpr_{diabetes}$ oli 59 prosenttia ja $tpr_{normaali}$ oli 86 prosenttia (taulukko 27). 20 ja 30 prosentin lisäyksillä neuroverkon antama tarkkuus pysyi lähes samana kuin alkuperäisessä aineistossa, mutta $tpr_{diabetes}$ - ja $tpr_{normaali}$ -arvoissa oli selkeä ero verrattuna alkuperäiseen aineistoon. Mitä enemmän tapauksia on lisätty *diabetes*-luokkaan, sitä suurempi on $tpr_{diabetes}$. Vastaavasti $tpr_{normaali}$ laskee kun *diabetes*-luokkaan lisätään tapauksia.

Taulukko 27. Pima-aineiston antamat tarkkuudet ja tpr -arvot eri aineistoilla (keskiarvo/mediaani).

Aineisto	Tarkkuus	$tpr_{diabetes}$	$tpr_{normaali}$
Alkuperäinen	77%/76%	59%/60%	86%/86%
20% lisäys	76%/76%	65%/66%	83%/83%
30% lisäys	76%/76%	69%/69%	82%/82%

Taulukon 27 tuloksia tarkastelemalla voidaan huomata mitä tapausten keinotekoinen lisääminen saa aikaiseksi Pima-aineiston luokitteluun neuroverkoilla. Tekemällä tilastollisia testejä voidaan tarkemmin tutkia onko tapausten keinotekoisella lisäämisellä todellisuudessa paljoa vaikutusta Pima-aineiston luokitteluun. Kun tehdään U-testi tarkkuuksille, niin saadaan taulukon 28 tulokset. Tämä vahvistaa taulukosta 27 nähtävää tulosta, jonka

mukaan tarkkuus ei juuri muutu, vaikka tapauksia lisättäisiin keinotekoisesti Pima-aineistoon. Kun samat testit tehtiin molempien luokkien *tpr*-arvoille, niin tulos oli aina nolla prosenttia, eli aineistojen *tpr*-arvot ovat erisuuret. Tämä vahvistaa sen, että tapausten keinotekoinen lisääminen Pima-aineistoon vaikuttaa molempien luokkien *tpr*-arvoihin.

Taulukko 28. U-testien tulokset tarkkuuksille Pima-aineistosta. Tulos kertoo kuinka suurella todennäköisyydellä kahdesta eri aineistosta saadut tarkkuudet ovat samat.

Aineisto	Tarkkuus
Alkuperäinen - 20% lisäys	71%
Alkuperäinen - 30% lisäys	17%
20% lisäys - 30% lisäys	69%

Pima-aineistossa tapausten keinotekoinen lisääminen tuotti hypoteesin mukaisen tuloksen, eli pienemmän luokan (*diabetes*) luokittelu parani. Samalla suuremman luokan (*normaali*) luokittelu heikkeni, joten tarkkuus ei juuri muuttunut yhtään. Myös U-testin perusteella tarkkuus oli lähes sama alkuperäiselle ja 20 prosentin lisäyksen aineistolle. Neuroverkon testiaineistoista antamien keskineliövirheiden keskiarvo oli jokaisella aineistolla n. 0,4, joten neuroverkko oppi jokaisen aineiston tässäkin suhteessa yhtä hyvin. Koska sekä tarkkuus että keskineliövirheen neliöjuuri pysyivät lähes samoina jokaisella aineistolla, ei tapausten keinotekoisella lisäämisellä tarkkuuden ja keskineliövirheiden neliöjuuren suhteen Pima-aineistolle ole juuri vaikutusta. Tätä tulosta vahvistaa taulukon 28 U-testin tulokset tarkkuuksille. Pima-aineiston tapauksessa *tpr*-arvojen nousu ja lasku ovat jopa suhteellisen lineaarisia.

Pima-aineistossa mielenkiinnon kohteena on diabetes-diagnoosi. Tutkittaessa tuloksia tästä näkökulmasta, huomataan että 20 ja 30 prosentin lisäyksillä diabetes havaittiin paremmin kuin alkuperäisellä aineistolla ($tpr_{diabetes}$). Aineistoilla, joihin on lisätty keinotekoisesti tapauksia, suurempi osa tapauksista luokitellaan luokkaan *diabetes* kuin alkuperäisellä aineistolla. Keinotekoisesti suurennetuilla aineistoilla mahdollinen diabetes pystytään havaitsemaan paremmin kuin alkuperäisellä aineistolla. Pima-aineistoa tutkittaessa tapausten keinotekoinen lisääminen parantaa toivottua tulosta,

vaikka tarkkuus ei paranekaan, sillä mahdollinen diabetes pystytään havaitsemaan paremmin.

5.3. Tapausten keinotekoinen lisääminen virtsainkontinenssi-aineistoon

Virtsainkontinenssi-aineiston luokkien absoluuttiset ja suhteelliset frekvenssit, sekä vinous alkuperäiselle ja kahdelle keinotekoisesti suurennetulle aineistolle näkyvät taulukossa 29. Tapausten keinotekoisella lisäämisellä saadaan suurimman luokan (*stress*) keskimääräistä suhteellisen frekvenssin eroa muihin luokkiin vähennettyä kymmenen prosenttiyksikköä. Tässä ei päästä niin lähelle tasaista jakaumaa kuin päästiin Pima- ja huimaus-aineistoissa. Luokkamuuttujan jakauma jää edelleen vinoksi. Alkuperäisen aineiston koko on 529 tapausta, 20 prosentin lisäyksellä aineiston koko on 570 tapausta ja 30 prosentin lisäyksellä aineiston koko on 591 tapausta. Virtsainkontinenssi-aineisto opetettiin neuroverkolle, jossa oli seitsemän piiloneuronia. Tämä piiloneuronien määrä katsottiin olevan optimaalinen virtsainkontinenssi-aineiston luokitteluun luvun 4.8 perusteella.

Taulukko 29. Virtsainkontinenssi-aineiston frekvenssit ja vinoudet alkuperäiselle aineistolle sekä 20 ja 30 prosenttia pienemmän luokan tapauksia lisättyihin aineistoihin.

Aineisto	Absoluuttinen frekvenssi (suhteellinen)			Vinous
	<i>muut</i>	<i>mixed</i>	<i>stress</i>	
Alkuperäinen	66 (12%)	140 (27%)	323 (61%)	1,01
20% lisäys	79 (14%)	168 (29%)	323 (57%)	0,85
30% lisäys	86 (15%)	182 (31%)	323 (54%)	0,78

Luokka *muut* on saatu yhdistämällä kolme pienempää luokkaa yhdeksi luokaksi. Tapausten keinotekoinen lisääminen kannattaisi tehdä *muut*-luokalle siten, että kutakin sen kolmesta luokasta lisättäisiin keinotekoisesti erikseen. Tässä tutkielmassa käytetyssä virtsainkontinenssi-aineistossa *muut*-luokka oli valmiiksi yhdistetty, joten siihen on lisätty keinotekoisesti tapauksia koko *muut*-luokan jakauman perusteella. Tällä tavalla keinotekoisesti lisätyt *muut*-luokan tapaukset saattavat olla erilaisia alkuperäisiin kolmeen luokkaan verrattuna.

Taulukossa 30 on virtsainkontinenssi-aineiston antamat tulokset eri aineistoilla kun tulosneuroneja oli yksi ja taulukossa 31 on tulokset kolmella tulosneuronilla (ks. luku 4.8). Virtsainkontinenssi-aineiston kohdalla tarkkuuksien keskiarvo on laskenut vähän kun tapauksia on lisätty aineistoon keinotekoisesti. Tarkkuuden muutos on hyvin pieni, eikä voida tehdä johtopäätöksiä tarkkuuden laskemisesta kun tapauksia lisätään aineistoon keinotekoisesti. Aikaisemmin tehty hypoteesi, että pienempien luokkien tpr -arvot nousevat ja suurimman luokan tpr -arvo laskee, ei pidä paikkaansa kaikille luokille virtsainkontinenssi-aineistossa. Yhden tulosneuronin neuroverkoilla tpr_{muut} on laskenut sitä enemmän mitä enemmän luokkaan *muut* on tapauksia keinotekoisesti lisätty. Nousu tpr_{mixed} -arvolle on vain 1-3 prosenttiyksikköä, joten nousu ei ole niin suuri kuin aikaisemmilla kahdella aineistolla. Vaikka muihin luokkiin lisättiin keinotekoisesti tapauksia, niin tpr_{stress} ei muuttunut juuri yhtään. Eräs syy tähän on mahdollisesti, että luokan *stress* absoluuttinen ja suhteellinen frekvenssi ovat edelleen hyvin suuria, vaikka muihin luokkiin on lisätty keinotekoisesti tapauksia.

Taulukko 30. Virtsainkontinenssi-aineiston antamat tarkkuudet ja tpr -arvot eri aineistoilla yhdellä tulosneuronilla (keskiarvo/mediaani).

Aineisto	Tarkkuus	tpr_{muut}	tpr_{mixed}	tpr_{stress}
Alkuperäinen	88%/88%	82%/86%	76%/77%	95%/95%
20% lisäys	87%/88%	78%/80%	77%/78%	94%/94%
30% lisäys	87%/88%	77%/78%	79%/79%	94%/94%

Luvussa 4.8 huomattiin, että kolmen tulosneuronin neuroverkko pystyi luokittelemaan paremmin luokkaa *mixed* ja huonommin luokkaa *muut*, kuin yhden tulosneuronin neuroverkko. Tilanne on sama, kun verrataan taulukkojen 30 ja 31 aineistojen eroja. Tapausten keinotekoinen lisääminen vaikutti tarkkuuteen ja tpr -arvoihin samalla tavalla: tpr_{muut} laski, tpr_{mixed} nousi ja tpr_{stress} ei muuttunut juuri yhtään. Tulosneuronien lukumäärällä ei siis ole tarkkuuksien keskiarvojen suhteen vaikutusta tapausten keinotekoisesta lisäämisestä kannalta virtsainkontinenssi-aineistoon.

Taulukko 31. Virtsainkontinenssi-aineiston antamat tarkkuudet ja *tpr*-arvot eri aineistoilla kolmella tulosneuronilla (keskiarvo/ mediaani).

Aineisto	Tarkkuus	<i>tpr</i> _{muut}	<i>tpr</i> _{mixed}	<i>tpr</i> _{stress}
Alkuperäinen	89%/90%	71%/75%	85%/86%	94%/94%
20% lisäys	88%/88%	72%/73%	85%/87%	93%/94%
30% lisäys	88%/88%	74%/75%	87%/89%	93%/93%

Tarkastellaan vielä neuroverkkojen antamia tarkkuuksia ja *tpr*-arvoja U-testillä. Taulukoiden 30 ja 31 perusteella merkittävä ero alkuperäisen ja keinotekoisesti suurennettun aineiston välillä on *tpr*_{muut}-arvossa ja loput arvot eivät ole paljoa muuttuneet. Taulukossa 32 on U-testin tulokset yhden tulosneuronin neuroverkoille ja taulukossa 33 kolmen tulosneuronin neuroverkoille. Yhden tulosneuronin neuroverkoilla mistään aineistoista saadut tulokset eivät ole samanlaisia. U-testin tuloksen mukaan yhdellä tulosneuronilla neuroverkkojen luokittelulla on selviä eroja alkuperäisen ja keinotekoisesti suurennettujen aineistojen välillä. Ainoastaan 20 ja 30 prosentin lisäyksillä on tuloksissa lievää samanlaisuutta. Eniten samanlaisia ovat tarkkuus ja *tpr*_{stress}, jotka taulukon 30 mukaan eivät myöskään muutu juuri yhtään 20 ja 30 prosentin lisäyksillä.

Taulukko 32. U-testien tulokset tarkkuuksille virtsainkontinenssi-aineistosta yhden tulosneuronin neuroverkolla. Tulos kertoo kuinka suurella todennäköisyydellä kahdesta eri aineistosta saadut tarkkuudet ja *tpr*-arvot ovat samat.

Aineisto	Tarkkuus	<i>tpr</i> _{muut}	<i>tpr</i> _{mixed}	<i>tpr</i> _{stress}
Alkuperäinen - 20% lisäys	0%	0%	11%	0%
Alkuperäinen - 30% lisäys	0%	0%	0%	0%
20% lisäys - 30% lisäys	17%	6%	0%	28%

Kolmen tulosneuronin neuroverkkojen tarkkuuksissa ja *tpr*-arvoissa alkuperäisen ja keinotekoisesti suurennettujen aineistojen välillä on

havaittavissa taulukon 33 mukaan enemmän samankaltaisuutta, kuin yhden tulosneuronin neuroverkoilla. Varsinkin alkuperäisen ja 20 prosentin lisäyksen aineistojen tpr_{muut} ja tpr_{mixed} ovat käytännössä samoja. Tämä kertoo, ettei luokkien *muut* ja *mixed* luokittelu muuttunut juuri yhtään, kun niihin lisättiin keinotekoisesti 20 prosenttia lisää tapauksia. Taulukon 31 perusteellakin tpr_{muut} ja tpr_{mixed} ovat lähes samoja näillä kahdella aineistolla. Koska sekä alkuperäisellä että 20 prosentin lisäyksen aineistoilla luokkien *muut* ja *mixed* luokittelu ei muutu ja tarkkuus ja tpr_{stress} huononevat vähän, niin 20 prosentin lisäyksen aineistolla ei saavuteta parannusta luokitteluun kolmen tulosneuronin neuroverkoilla.

Taulukko 33. U-testien tulokset tarkkuuksille virtsainkontinenssi-aineistosta kolmen tulosneuronin neuroverkolla. Tulos kertoo kuinka suurella todennäköisyydellä kahdesta eri aineistosta saadut tarkkuudet ja tpr -arvot ovat samat.

Aineisto	Tarkkuus	tpr_{muut}	tpr_{mixed}	tpr_{stress}
Alkuperäinen - 20% lisäys	0%	82%	77%	0%
Alkuperäinen - 30% lisäys	0%	20%	0%	0%
20% lisäys - 30% lisäys	0%	22%	0%	5%

Kolmen tulosneuronin neuroverkoilla 30 prosentin lisäyksen aineiston tarkkuus ja tpr -arvot eivät U-testin (taulukko 33) perusteella ole samoja. Ainoastaan tpr_{muut} on vähän samanlainen. Kun vertaa taulukon 31 alkuperäisen ja 30 prosentin lisäyksen aineistojen tarkkuuksia ja tpr -arvoja, niin voidaan huomata että tpr_{muut} ja tpr_{mixed} ovat paremmat keinotekoisesti suurennettulla aineistolla. Näiden arvojen eroavuuden vahvistaa taulukossa 33 olevat U-testin tulokset. Näiden tulosten perusteella 30 prosentin lisäyksen aineisto luokittelee virtsainkontinenssi-aineiston eri tavalla kuin alkuperäinen aineisto kolmen tulosneuronin neuroverkoilla.

Virtsainkontinenssi-aineistolla tapausten keinotekoisella lisäämisellä oli negatiivinen vaikutus niin tarkkuuteen kuin tpr_{stress} -arvoihinkin sekä yhden että kolmen tulosneuronin neuroverkoilla. Suurimman luokan tpr -arvo ei muuttunut juuri lainkaan, kun aineistoon lisättiin keinotekoisesti tapauksia. Yhden tulosneuronin neuroverkoilla oli neuroverkkojen tarkkuuksilla ja tpr -arvoilla eroa alkuperäisen ja keinotekoisesti suurennettujen aineistojen välillä.

Mitä enemmän tapauksia lisättiin keinotekoisesti aineistoon, niin tpr_{muut} laski merkittävästi, joten yhden tulosneuronin neuroverkoilla ei saavutettu mitään hyötyä tapausten keinotekoisella lisäämisellä. Kolmen tulosneuronin neuroverkoilla ei tullut aineiston luokitteluun parannusta alkuperäisen ja 20 prosentin lisäyksen aineistojen välillä. Alkuperäisen ja 30 prosentin lisäyksen aineistojen välisissä tuloksissa taas oli eroja. Tarkkuus ja tpr_{stress} laskivat vähän, mutta tpr_{muut} ja tpr_{mixed} nousivat. Jos mielenkiinnon kohteena on luokka *muut* tai *mixed*, niin kolmen tulosneuronin neuroverkoilla niiden luokkien tapausten keinotekoinen lisääminen 30 prosentilla tuottaisi näiden luokkien kannalta parempia tuloksia.

5.4. Yhteenveto tapausten keinotekoisesta lisäämisestä

Edellä lisättiin huimaus-, Pima- ja virtsainkontinenssi-aineistoihin keinotekoisesti tapauksia ja verrattiin miten neuroverkko oppii ne alkuperäisiin aineistoihin verrattuna. Luvun alussa tehtiin hypoteesi, miten tapausten keinotekoinen lisääminen aineistoihin vaikuttaa neuroverkon luokitteluun: tapausten keinotekoinen lisääminen ja sitä kautta luokkien jakauman tasoittaminen parantaa pienempien luokkien luokittelua ja tpr -arvoja, ja vastaavasti huonontaa suurimman luokan luokittelua ja tpr -arvoa.

Huimaus- ja Pima-aineisto käyttäytyivät tapausten keinotekoisesta lisäämisestä jälkeen hypoteesin mukaan. Molemmissa pienempien luokkien, eli luokkien, joihin lisättiin keinotekoisesti tapauksia, tpr -arvot paranivat. Vastaavasti suurimman luokan tpr -arvo laski. Kummassakaan aineistossa tarkkuus ei kuitenkaan muuttunut merkittävästi, koska toisten luokkien tpr -arvot nousivat samalla, kun muiden luokkien tpr -arvot nousivat.

Virtsainkontinenssi-aineiston kohdalla taas tarkkuuksien ja tpr -arvojen muutos riippui myös tulosneuronien lukumäärästä. Kolmella tulosneuronilla alkuperäisen ja keinotekoisesti suurennettujen aineistojen tarkkuuksien ja tpr -arvojen ero oli pienempi kuin yhdellä tulosneuronilla. Virtsainkontinenssi-aineiston kohdalla 30 prosentin lisäyksen aineistoa kannattaa käyttää, jos kiinnostuksen kohteena on luokka *muut* tai *mixed*. Muussa tapauksessa tapausten keinotekoinen lisääminen ei tuota muutosta tai huonontaa luokittelua.

Näiden tulosten perusteella tapausten keinotekoinen lisääminen aineiston pienimpiin luokkiin parantaa yleensä pienempien luokkien luokittelua ja huonontaa suurempien luokkien luokittelua käytettäessä neuroverkkoja. Neuroverkkojen antama tarkkuuksien keskiarvo pysyy yleensä suurin piirtein samana tapausten keinotekoisesta lisäämisestä jälkeen, koska samalla kun toisten luokkien luokittelu paranee, niin toisten luokkien luokittelu huononee.

6. Yhteenveto

Tutkielmassa tarkasteltiin neuroverkkoja luokittelijoina ja miten tapausten keinotekoinen lisääminen aineistoon vaikuttaa neuroverkkojen oppimiseen. Tutkielmassa käytettiin yhteensä kahdeksaa eri aineistoa, joista kolmella tutkittiin miten tapausten keinotekoinen lisääminen vaikuttaa neuroverkkojen oppimiseen. Aineistoista viisi oli peräisin UCI:sta ([Blake and Merz 1998]) ja kolme Tampereen yliopiston tietojenkäsittelytieteiden laitokselta.

Kahdeksan eri aineiston opettamisella neuroverkoille oli tarkoitus tutkia neuroverkkojen oppimiskykyä. Lisäksi tarkoituksena oli tutkia mitkä asiat mahdollisesti vaikuttavat neuroverkkojen oppimiseen. Neuroverkkojen kykyä luokitella tarkasteltiin tarkkuuksien, *tpr*-arvojen ja keskineliövirheiden perusteella. Selkein tulos oli se, että neuroverkko oppi aina suurimman luokan parhaiten ja usein pienimmän luokan huonoiten. Tulokset havainnollistivat myös neuroverkkojen luokittelumahdollisuutta piiloneuronien lukumäärien suhteen. Neuroverkko pystyi luokittelemaan korkeintaan piiloneuronien lukumäärän luokkia. Aineistojen opettamiseen neuroverkoille käytettiin kymmenenkertaista ristiinvalidointia. Lisäksi jokaisella ristiinvalidointikerralla neuroverkko opetettiin kymmenen kertaa, jotta välttyttäisiin paikallisilta minimeiltä (kuva 13). Neuroverkolle opetettiin siis aineisto yhteensä 100 kertaa ja näiden opetusten tarkkuuksien ja *tpr*-arvojen keskiarvoja ja mediaaneja tutkittiin. Jokaiselle aineistolle etsittiin mahdollisimman hyvä arkkitehtuuri. Joidenkin arkkitehtuurien antamien tulosten ero oli niin pieni, ettei selvästi voinut sanoa mikä olisi ollut paras arkkitehtuuri kyseiselle aineistolle.

Alokas-aineisto koostui Ruotsin armeijan kutsunnoissa olleiden kuulotutkimuksista. Alokas-aineistoa neuroverkko ei pystynyt käytännössä oppimaan yhtään. Neuroverkko pääsi keskimäärin 53 prosentin tarkkuuteen ja tpr_{normaali} oli 94 prosenttia ja tpr_{huono} oli 14 prosenttia. Neuroverkko luokitteli siis lähes kaikki tapaukset luokkaan *normaali*. Saatu 53 prosentin tarkkuuskin on lähes sama mikä saataisiin arvaamalla jokainen tapaus satunnaisesti.

Buba-aineisto käsitteli potilaita, joilla oli maksan häiriöitä. Potilaat oli diagnosoitu joko terveiksi tai sairaksi. Neuroverkko pystyi luokittelemaan Buba-aineiston kolmella piiloneuronilla 71 prosentin tarkkuudella, tpr_{terve} oli n. 80 prosenttia ja tpr_{sairas} oli n. 60 prosenttia. Aineistossa oli 200 tervettä ja 145 sairasta, joten neuroverkko oppi paremmin suuremman luokan.

Ecoli-aineistossa on tarkasteltu proteiinin muodostumispaikkoja. Neuroverkko pystyi luokittelemaan ecoli-aineiston keskimäärin 88 prosentin tarkkuudella, kun Blake ja Merz [1998] ovat saaneet aikaisemmin luokiteltua aineiston 81 prosentin tarkkuudella binäärisillä päätöspuilla ja Bayesin

luokittelulla. Neuroverkkojen antama tarkkuus oli siis parempi. Neuroverkko oppi *ecoli*-aineistostakin parhaiten suurimman luokan, kun tpr_{cp} oli koko ajan lähes 100 prosenttia.

Haberman-aineisto käsitteli potilaiden selviytymistä rintasyöpäleikkauksista. Tarkkuudeksi saatiin 73 prosenttia, mikä on sama kuin jos kaikki tapaukset arvattaisiin selvinneen leikkauksesta hengissä yli viisi vuotta (luokka *selviytyi*). Testiaineistoista saatu tpr_{kuoli} oli n. 30 prosenttia ja $tpr_{selviytyi}$ oli n. 88 prosenttia, joten tässäkin tapauksessa neuroverkko oppi paremmin suuremman luokan. Luokan *selviytyi* suhteellinen frekvenssi oli peräti 73 prosenttia.

Huimaus-aineisto käsitteli erilaisia huimaussairauksia, joita oli aineistossa kuusi. Neuroverkon antama tarkkuus oli 80 prosenttia, mitä voidaan pitää hyvänä tuloksena. Neuroverkko oppi huimaus-aineistosta kaikki luokat hyvin, paitsi luokkaa *sd* se ei oppinut. Kaikki tpr -arvot olivat yli 70 prosenttia, paitsi tpr_{sd} oli vain 20 prosenttia. Luokka *sd* oli pienin luokka ja sen suhteellinen frekvenssi oli vain viisi prosenttia, mikä mahdollisesti vaikeutti sen oppimista.

New-thyroid -aineisto käsitteli kilpirauhasen toimintaa. Neuroverkko pystyi luokittelemaan sen 96 prosentin tarkkuudella. Kaikki tpr -arvotkin olivat yli 90 prosenttia. Luokan *normaali*, joka oli suurin luokka, neuroverkko pystyi luokittelemaan lähes 100 prosentin tarkkuudella, kun $tpr_{normaali}$ oli keskimäärin 98 prosenttia ja mediaani 100 prosenttia.

Pima-aineisto käsitteli Pima-intiaaniheimon naisten diabetesta. Luokkamuuttuja kertoi oliko ko. tapauksella diabetes vai ei. Neuroverkko pystyi luokittelemaan Pima-aineiston 76 prosentin tarkkuudella ja $tpr_{diabetes}$ oli 59 prosenttia ja $tpr_{normaali}$ oli 86 prosenttia. Pima-aineistollakin neuroverkko oppi paremmin suuremman luokan, kun luokan *normaali* suhteellinen frekvenssi oli 65 prosenttia.

Virtsainkontinenssi-aineisto käsitteli tahattomasta virtsankarkailusta kärsivien naisten potilastietoja. Neuroverkko pystyi luokittelemaan virtsainkontinenssi-aineiston 88 prosentin tarkkuudella. Neuroverkko oppi parhaiten suurimman luokan antaen tpr_{stress} -arvoksi 95 prosenttia. Luokan *stress* suhteellinen frekvenssi oli 61 prosenttia ja aineistossa oli yhteensä kolme luokkaa.

Luvussa 5 lisättiin huimaus-, Pima- ja virtsainkontinenssi-aineistojen pienimpiin luokkiin 20 ja 30 prosenttia uusia tapauksia keinotekoisesti ja tutkittiin miten se vaikuttaa neuroverkoilla luokitteluun. Tässä käytettiin kymmenen kertaa kymmenkertaista ristiinvalidointia. Pienempien luokkien tapauksien keinotekoisien lisäämisen vaikutuksesta luokitteluun tehtiin hypoteesi luvun 4 tulosten perusteella: tapauksien keinotekoinen lisääminen ja

sitä kautta luokkien jakauman tasoittaminen parantaa pienempien luokkien luokittelua ja *tpr*-arvoja ja vastaavasti huonontaa suurimman luokan luokittelua ja *tpr*-arvoa. Huimaus- ja Pima-aineistojen kohdalla tuloksen olivat hypoteesin mukaisia, mutta virtsainkontinenssi-aineiston kohdalla muutos oli vähän erilainen. Tapausten keinotekoinen lisääminen ei muuttanut neuroverkkojen antamaa tarkkuutta merkittävästi, koska samalla kun toisten luokkien *tpr*-arvot nousivat, niin toisten luokkien *tpr*-arvot laskivat. Yleensä tarkkuuksien keskiarvo laski noin yhden prosenttiyksikön.

Yleisesti tapauksia kannattaa lisätä keinotekoisesti kiinnostuksen kohteena oleviin luokkiin, sillä näin niiden *tpr*-arvoja saadaan nostettua. Tällöin kuitenkin muiden luokkien *tpr*-arvot laskevat. Koska toisten luokkien *tpr*-arvot laskevat ja toisten nousevat, ei tapausten keinotekoisella lisäämisellä saada paljoa muutosta tarkkuuteen. Yleensä tarkkuus laskee vähän. Se miten tapausten keinotekoinen lisääminen aineistoon vaikuttaa neuroverkkojen luokitteluun, on tässä tutkielmassa saatujen tulosten perusteella tapauskohtaista ja riippuu kiinnostuksen kohteena olevasta luokasta, eikä neuroverkko tarkkuuden suhteen pysty luokitteluun keinotekoisesti suurennettua aineistoa paremmin kuin alkuperäistä aineistoa.

Viiteluettelo

- [Astion and Wilding, 1992] Michael L. Astion and Peter Wilding, The Application of Backpropagation Neural Networks to Problems in Pathology and Laboratory Medicine. *Archives of Pathology & Laboratory Medicine* **116** (1992), 995-1001.
- [Auramo, 1999] Yrjö Auramo, *Construction of an Expert System to Support Otoneurological Vertigo Diagnosis*, Report A-1999-2. Department of Computer and Information Sciences, University of Tampere, 1999.
- [Baxt, 1995] William G. Baxt, Application of Artificial Neural Networks to Clinical Medicine. *Lancet* **346** (1995), 1135-1138.
- [Baxt, 1991] William G. Baxt, Use of Artificial Neural Network for the Diagnosis of Myocardial Infarction. *Annals of Internal Medicine* **115** (1991), 843-848.
- [Bishop, 1995] Christopher M. Bishop, *Neural Networks for Pattern Recognition*. Clarendon Press, 1995.
- [Blake and Merz, 1998] C.L. Blake and C.J. Merz, UCI Repository of machine learning databases. Irvine, University of California, Department of Information and Computer Science. Available as <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
- [Callan, 1999] Robert Callan, *The Essence of Neural Networks*. Prentice Hall Europe, 1999.
- [Fu, 1994] LiMin Fu, *Neural Networks in Computer Intelligence*. McGraw-Hill International Editions, 1994.
- [Guerriere and Detsky, 1991] Michael R. J. Guerriere and Allan S. Detsky, Neural Networks: What Are They? *Annals of Internal Medicine* **115** (1991), 906-907.
- [Haykin, 1994] Simon Haykin, *Neural Networks a Comprehensive Foundation*. Prentice Hall International Editions, 1994.
- [Hecht-Nielsen, 1990] Robert Hecht-Nielsen, *Neurocomputing*. Addison-Wesley Publishing Company, 1990.
- [Jerez-Aragonés *et al.*, 2003] José M. Jerez-Aragonés, José A. Gómez-Ruiz, Gonzalo Ramos-Jiménez, José Muñoz-Pérez and Emilio Alba-Conejo, A Combined Neural Network and Decision Trees Model for Prognosis of Breast Cancer Relapse. *Artificial Intelligence in Medicine* **27** (2003), 45-63.
- [Juhola, 2000] Martti Juhola, *Neurolaskenta*. Kurssimateriaali, Tietojenkäsittelytieteiden laitos, Tampereen yliopisto, 2000.

- [Karapataki and De Wilde, 1997] M. Karapataki and P. De Wilde, Hopfield Network Applied to Blood Vessel Detection in Angiograms. *Medical & Biological Engineering & Computing* **35** (1997), 428-430.
- [Kennedy *et al.*, 1997] R.L. Kennedy, R.F. Harrison, A.M. Burton, H.S. Fraser, W.G. Hamer, D. MacArthur, R. McAllum and D.J. Steedman, An Artificial Neural Network System for Diagnosis of Acute Myocardial Infarction (AMI) in the Accident and Emergency Department: Evaluation and Comparison with Serum Myoglobin Measurements. *Computer Methods and Programs in Biomedicine* **52** (1997), 93-103.
- [Laurikkala, 2001a] Jorma Laurikkala, *Knowledge Discovery for Female Urinary Incontinence Expert System*. Report A-2001-6. Department of Computer and Information Sciences, University of Tampere, 2001.
- [Laurikkala *et al.*, 2001b] Jorma Laurikkala, Martti Juhola, Seppo Lammi, Jorma Penttinen and Pauliina Aukee, Analysis of the Imputed Female Urinary Incontinence Data for the Evaluation of Expert System Parameters. *Computers in Biology and Medicine* **31** (2001), 239-257.
- [Looney, 1997] Carl G. Looney, *Pattern Recognition Using Neural Networks*. Oxford University Press, 1997.
- [Maclin and Dempsey, 1992] Philip S. Maclin and Jack Dempsey, Using an Artificial Neural Network to Diagnose Hepatic Masses. *Journal of Medical Systems* **16** (1992), 215-225.
- [Maclin *et al.*, 1991] Philip S. Maclin, Jack Dempsey, Jay Brooks and John Rand, Using Neural Networks to Diagnose Cancer. *Journal of Medical Systems* **15** (1991), 11-19.
- [MATLAB, 2002] The Mathworks Inc, *MATLAB Help*. The Mathworks Inc., 2002.
- [Pett, 1997] M. A. Pett, *Nonparametric Statistics for Health Care Research: Statistics for Small Samples and Unusual Distributions*. SAGE Publications, 1997.
- [Picton, 2000] Phil Picton, *Neural Networks*. Palgrave, 2000.
- [Rojas, 1996] Raul Rojas, *Neural Networks, a Systematic Introduction*. Springer, 1996.
- [Swingler, 1996] Kevin Swingler, *Applying Neural Networks, a Practical Guide*. Academic Press, 1996.
- [Viikki, 2003] Kati Viikki, *Tietämyksen muodostaminen*. Kurssimateriaali, Tietojenkäsittelytieteiden laitos, Tampereen yliopisto, 2003.
- [Viikki, 2002] Kati Viikki, *Machine Learning on Otoneurological Data: Decision Trees for Vertigo Diseases*. Report A-2002-8. Department of Computer and Information Sciences, University of Tampere, 2002.

- [Viikki *et al.*, 2001] Kati Viikki, Martti Juhola, Ilmari Pyykkö and Pekka Honkavaara, Evaluating Training Data Suitability for Decision Tree Induction. *Journal of Medical Systems* **25** (2001), 133-144.
- [Viikki *et al.*, 2000] Kati Viikki, Erna Kentala, Martti Juhola and Ilmari Pyykkö, Confounding Values in Decision Trees Constructed for Six Otoneurological Diseases. In: *Proceedings of the Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP-2000)*, 58-60.
- [Viikki *et al.*, 1999] Kati Viikki, Erna Kentala, Martti Juhola and Ilmari Pyykkö, Decision Tree Induction in the Diagnosis of Otoneurological Diseases. *Medical Informatics & the Internet in Medicine* **24** (1999), 277-289.
- [Wu and Yen, 1992] F. Y. Wu and K. K. Yen, A PC-Based Neural Network for On-Line Measurement of Regional Cerebral Blood Flow. *Computers in Biology and Medicine* **22** (1992), 23-32.