

Itseorganisoituva kartta sekä sen laajennosten eroavaisuudet

TURUN YLIOPISTO
Tietotekniikan laitos
LuK-tutkielma
Tietojenkäsittelytiede
Huhtikuu 2024
Henri Kartano

TURUN YLIOPISTO
Tietotekniikan laitos

HENRI KARTANO: Itseorganisoituva kartta sekä sen laajennosten eroavaisuudet

LuK-tutkielma, 24 s.
Tietojenkäsittelytiede
Huhtikuu 2024

Tämä tutkielma on kirjallisuuskatsaus itseorganisoituvan kartan (self-organizing map, SOM) ominaisuuksiin ja sovelluksiin, sekä alkuperäisen SOMin pohjalta kehitettyjen muunnelmien ja laajennosten ominaisuuksien, erojen ja käyttötarkoitusten vertailu sekä alkuperäiseen SOMiin, että toisiinsa. Katsauksen tarkoituksena on aluksi selvittää SOMin algoritmin toiminta, käyttötarkoitus ja rajoitteet, jonka jälkeen laajentaa selvitys monimutkaisemmin toimiviin alkuperäisen SOMin algoritmin laajennettuihin muunnelmiin. Katsauksessa selvennetään myös uudempia, 2010-luvun jälkeen julkaistuja vähemmän tunnettuja muunnelmia, tunnetuimpien muunnosten ja laajennosten ollessa useita kymmeniä vuosia vanhoja. Tutkielmassa selvitetään myös syitä laajennosten kehittämiseksi, kartoittaen alkuperäisen SOMin heikkouksia ja vahvuuksia.

Asiasanat: itseorganisoituva kartta, SOM, neuroverkot, ohjaamaton oppiminen

Sisällys

1	Johdanto	1
2	Taustaa	4
3	Itseorganisoituva kartta, SOM	9
3.1	Algoritmi	10
3.2	Havainnollistus	12
4	Muunnelmat ja laajennokset	16
4.1	Growing SOM, GSOM	17
4.2	Evolving SOM, ESOM	19
4.3	FlowSOM	20
4.4	NOASSOM	21
4.5	Muut laajennokset	22
5	Yhteenveto	23
	Lähdeluettelo	25

Kuvat

2.1	Esimerkki yksinkertaisesta neuroverkosta	4
2.2	Esimerkki itseorganisoituvasta kartasta	6
3.1	Esimerkki kouluttamattomasta kartasta.	14
3.2	Esimerkki koulutetusta kartasta.	14

1 Johdanto

Itseorganisoituva kartta (self organizing map, SOM) on koneoppimisen malli moniulotteisen datajoukon esittämiseen, jonka tarkoituksena on vähentää datajoukon ulottuvuuksien määrää ja siten muodostaa datajoukkoa havainnollistava helpommin ymmärrettävä abstraktio. Olennaista itseorganisoituvan kartan muodostamalle abstraktiolle on säilyttää syötedatan topologiset ja metriset suhteet. Ensimmäisen itseorganisoituvan kartan esitteli professori Teuvo Kohonen 1980-luvulla. Sittenmin menetelmää on sovellettu laajasti erilaisiin tutkimuskohteisiin muun muassa finanssialoilla osakekurssien ja rahoituslaitosten konkurssien ennustamiseen, luonnontieteissä sääilmiöiden mallintamiseen ja ryvästyksen, sekä lingvistiikassa puheentunnistukseen. [1, 2]

Tämä tutkielma käsittelee itseorganisoituvan kartan muodostamista tarkastelemalla sen algoritmia ja muokkaamista parempien abstraktointituloksien saavuttamiseksi erilaisille sovelluskohteille. Aiheen ymmärtämiseen vaadittavia termejä, kuten *neuroverkko*, *ohjaamaton oppiminen*, *ryvästyksen* ja *kilpaileva oppiminen*, tullaan selvittämään. Tutkielmassa käytetään käsiteltävien termien lyhenteitä, kuten edellä mainittu SOM, ja suomenkielisten käännösten puuttuessa termeistä puhutaan niiden englanninkielisillä vastineilla.

Vaikka SOM on teknologisesti jokseenkin iäkäs menetelmä, sen lukuisat implementaatiokohteet kertovat laajasta potentiaalista ja muuntautuvuuskyvystä. Alunperin laajimmillaan noin tuhannen neuronin verkot ovat laajentuneet laskentatehon

lisääntyessä miljoonan neuronin verkoiksi, ja vastaavasti myös syötedatajoukkojen koot ovat kasvaneet tuhansista useisiin miljooniin alkioihin, ollen samalla moniulotteisempia. [1, 3] Neljän vuosikymmenen aikana SOM:sta on muodostettu useita kymmeniä eri muunnelmia ja laajennoksia eri sovelluskohteisiin. Tutkielmassa näistä laajennoksista tullaan käsittelemään vain murto-osa. Tutkielmassa käsiteltäviksi laajennoksiksi valikoitiin laajennoksia, joista löytyy sekä kosolti laaja-alaista tutkimustietoa, että tulevaisuuden laajennuspotentiaalia. Useasta tutkielmaan valikoidusta laajennoksesta löytyykin jo edelleen laajennettuja muotoja.

Tämä tutkielma keskittyy tutkimaan kahta SOMeihin liittyvää aihepiiriä:

1: Self-organizing map, SOM

2: SOMin laajennokset ja niiden käyttötarkoitukset

Tutkielman ensimmäinen aihe keskittyy tutkimaan alkuperäistä Kohosen julkaisemaa menetelmää, sen algoritmiikkaan ja suunniteltuihin implementaatiokohteisiin sekä havainnollistamaan miksi SOMista on muodostunut potentiaalinen koneoppimismalli. Toisessa aiheessa esitellään esitellään SOMin muunnelmia ja laajennoksia. Näitä variaatioita verrataan paitsi alkuperäiseen SOMiin, myös toisiinsa havainnollistaen kyseisen variaation kehityksen syytä.

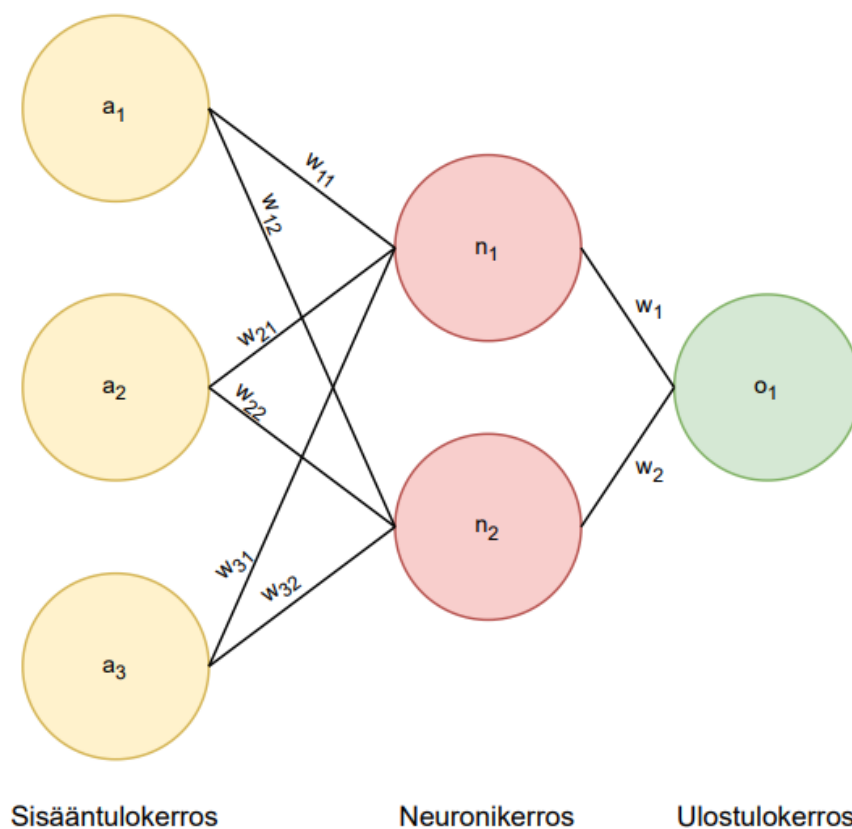
Tiedonhaku

Tutkielman lähdemateriaalin hakuun käytettiin pääasiassa Google Scholaria. Valtaosa lähteistä on saatavilla IEEE:n julkaisuportaalista. Lähteiden haku muodostettiin käyttämällä hakuoperaatioita, jotka sisältävät etsittävän algoritmin nimen tai akronyymin sekä mahdollisia avainsanoja. Lainausmerkkeillä ja binäärioperaatioilla voitiin painottaa jotakin tiettyä termiä. Uusimpien teknologioiden kohdalla

julkaisuvuosi voitiin rajata noin kymmeneen viime vuoteen. Monet uudemmat artikkelit viittaavat edelleen SOMin alkuperäiseen Kohosen julkaisemaan työhön. Tutkielmaan valittiin artikkeleita, jotka kuvaavat käsiteltävän algoritmin toimintaa tai käsittelevät sen vertailua toisiin malleihin.

2 Taustaa

Ennen tutkimuskysymyksiin perehtymistä on syytä selvittää aihepiiriin olennaisesti liittyviä termejä. Termejä käsitellään yleisellä tasolla, mutta tarpeen vaatiessa niiden merkitystä SOMien kontekstissa selvennetään.



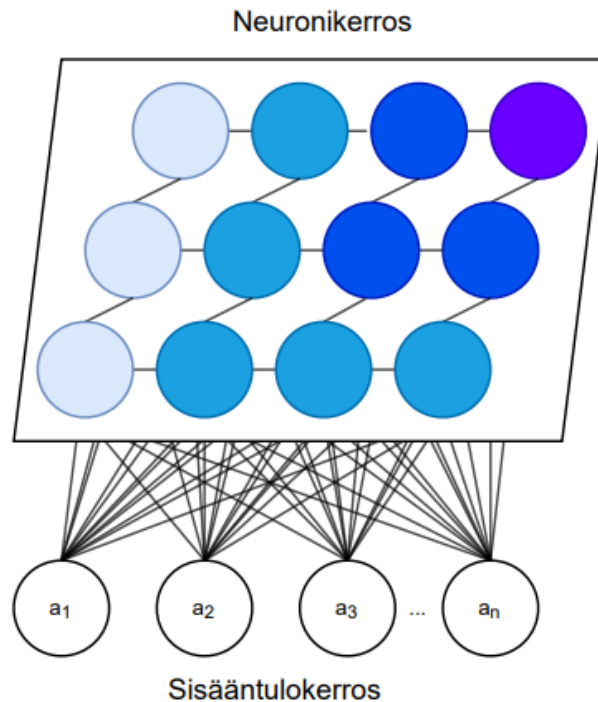
Kuva 2.1: Yksinkertainen neuroverkko, jossa a_i on syötealkio, n_i keinotekoinen piilotettu neuronin ja o_i ulostuloneuronin. Painovektorit w_{ij} syötealkion ja neuronin välillä ja vektorit w_i neuronin ja ulostuloneuronin välillä.

Keinotekoisella neuroverkolla (engl. Artificial Neural Network, ANN) tarkoitetaan biologisten neuroverkkojen innoittamaa laskentamallia, joka koostuu päällekkäisistä kerroksista: syötekerroksesta (input layer), yhdestä tai useammasta neuronikerroksesta (hidden layer) ja ulostulokerroksesta (output layer), kuten kuvassa 2.1. Näistä ensimmäinen eli syötekerros on informaatioalkioiden muodostama joukko, joka toimii syötteenä keinotekoisille neuroneille. Keinotekoiset neuronit muodostavat neuronikerroksen, joita voi karkeasti verrata biologisiin neuroneihin, joskaan keinotekoisien neuronien neuroverkot eivät pyri tarkasti noudattamaan biologisia orgaanisia neuroverkkoja. Neuronin saa syötteekseen koko syötekerroksen tai sen osajoukon. Neuronilla on tietty *painotus*, jonka perusteella syötealkion ja neuronin välille syntyy yhteys. Painotuksella tarkoitetaan neuronin vastaavuutta alkion tiettyyn ominaisuuteen, ulottuvuuteen. Tavanomaisessa neuroverkossa neuronin summaa alkioihin muodostamansa painotetut yhteydet ja välittää summan lineaarisen tai epälineaarisen aktivaatiofunktion läpi syötteeksi seuraavalle neuronikerrokselle tai ulostulokerrokselle. Tyypillisesti eri neuronikerrokset käyttävät samaa aktivaatiofunktiota. Ulostulokerroksen tarkoitus on koota neuronien sille syöttämät aktivaatiofunktion tulokset yhteen ja muodostaa siten haluttu lopputulos. [4]

SOMin muodostama neuroverkko (kuva 2.2) eroaa tavanomaisesta neuroverkosta siten, että SOMissa kerrostyyppejä on vain kaksi: neuronikerros on samalla ulostulokerros. SOMin neuronit ovat muiden ANNien tavoin yhteydessä kaikkiin syötealkioihin, mutta vain suurimman painotuksen yhteydellä on merkitystä; se saa arvon yksi, muut yhteydet käsitellään nollana. Tämä toimii samalla neuronin aktivaatiofunktiona, jota SOMien tapauksessa kutsutaan kilpailufunktioksi. [5]

Ohjaamaton oppiminen

Ohjaamattomasti oppiva (unsupervised learning) malli kykenee syötettä itsenäisesti iteroimalla tuottamaan haluttuja tuloksia syötteestä. Syötteen käsittelemättömyys



Kuva 2.2: Itseorganisoituva kartta, jossa syötealkiot a_n ja värítettö neuronikerros. Neuronikerroksen väreillä havainnollistetaan samankaltaisten syötealkioiden ryväs- tystä.

on eräs ohjaamattoman ja ohjatun oppimisen (supervised learning) ensisijaisista eroavaisuuksista - ohjattu oppiminen tarvitsee ulkopuolisen tahon leimaamaa dataa tuottaakseen oikeellista tulosta, kun vastaavasti ohjaamaton oppiminen muodostaa itsenäisesti tuloksia leimaamattomasta informaatiosta. Vastaavasti ohjatussa oppimisessä voidaan suorittaa virheenkorjausta esimerkiksi takaisinpropagaation (back-propagation) avulla, kun taas ohjaamattomassa oppimisessä hyödynnetään esimerkiksi Boltzmannin oppimista, suurimman uskottavuuden estimaattia tai niin kut- suttuja oppimissääntöjä, kuten Hebbianin oppimissääntö (Hebbian learning rule). SOMin tapauksessa oppiminen tapahtuu kilpailevan oppimisen kautta. Ohjatun ja ohjaamattoman oppimisen malleille annettuja tehtäviä on tyypillisesti jaoteltu si- ten, että ohjatun oppimisen tehtävät ovat luonteeltaan enemmän diskriminatiivisia, kuten esimerkiksi kuvan- ja hahmontunnistus, ja ohjaamattoman oppimisen tehtä-

vät luonteeltaan kategorisoivia, kuten datan kompressointi, ryvästys ja informaation abstrahointi. [4] [5]

Kilpaileva oppiminen

Kilpaileva oppiminen (competitive learning) on ohjaamattoman koneoppimisen (unsupervised learning) muoto, jossa algoritmi kykenee itseoppivasti hyödyntämään leimaamatonta dataa muodostaakseen haluttuja tuloksia. Erona ohjattuun oppimiseen (supervised learning) on juurikin syötedatan muoto - ohjattu oppiminen luottaa ulkopuolisen tahon leimaaman datan oikeellisuuteen. Kilpailevan oppimisen ideana on neuronien välinen *kilpailu*, jossa syötevektoreita vertaillaan yksi kerrallaan kaikkiin neuroneihin, joista yksi valitaan voittajaksi (best matching unit, BMU). Voittajan valintaan on olemassa useita tapoja. SOMin tapauksessa voittaja valitaan syötevektorin ja neuronin painovektorin euklidisen etäisyyden perusteella. Tavanomaisesti kilpailevassa oppimisessa voittajaneuronin löydyttyä vain voittajaneuronin painovektoria mukautetaan syötevektorin kaltaiseksi, mutta SOMin tapauksessa myös muiden kuin voittajaneuronin painovektoreita päivitetään lähemmäksi syötettä. SOMien luonteelle fundamentaalinen *naapurustofunktio* määrittää naapuruston koon, sekä valitun naapuruston neuroneiden muutoksen intensiteetin. Päivitysfunktio vetää voittajaneuronia ja sen naapurustoa syötevektoria kohti. Naapurustofunktio voi noudattaa erilaisia funktioita, yleisiä valintoja ovat esimerkiksi kuplafunktio (bubble function) tai Gaussin käyrä. Useimmiten naapurusto- ja päivitysfunktion vaikutus pienenee iteraatioiden edetessä. Tämän kaltainen toiminnallisuus on perusta SOMin nimessä esiintyvälle itseorganisoituvuudelle. [2, 4]

Ryvästys

Ryvästyksellä (klusterointi, clustering) tarkoitetaan joukon alkioiden ryhmittelyä alkioiden yhden tai useamman ominaisuuden mukaan. Ryvästyksen tärkeimpinä pää-

määrinä pidetään suuren datajoukon esittämistä tehokkaasti sekä datajoukon *ko-*
hinan eli vähemmän merkityksellisen informaation vähentäminen tarkemman ab-
straktion muodostamiseksi. [6] Ryvästysalgoritmeja on lukuisia, kuten esimerkiksi
 k :n keskiarvon klusterointimenetelmä (*k*-means clustering). SOMia ei itsessään pi-
detä ryvästysalgoritmina, sillä varsinainen ryvästys suoritetaan muodostetulle SO-
Mille. [7]

3 Itseorganisoituva kartta, SOM

SOM on myötäkytketty (engl. feedforward) neuroverkko, jonka itseorganisoituvuus perustuu ohjaamattoman kilpailevan oppimisen harjoitusalgoritmeihin, joiden tarkoituksena on järjestää ulostulokerroksen neuronit syötteen alkuperäisen topologisuuden mukaan. Ulostuloneuronit on useimmiten järjestetty kaksiulotteiseksi $m_1 \times m_2$ neuroneista koostuvaksi kartaksi. Neuronien paikat kartassa voi mieltää pysyvän samoina niiden painovektorien arvojen muuttuessa, jolloin syötteen ulottuvuuksia on vähennetty helpommin visualisoitavaan muotoon. Useamman ulottuvuuden ulostulokartat ovat myös mahdollisia, joskin abstraktion helppolukuisuuden kustannuksella. Syötteen topologian säilyttäminen saavutetaan järjestämällä syötevaruudessa toisiaan muistuttavat syötevektorit ulostuloneuroneihin, joiden painovektoreiden arvoa säädetään vastaamaan paremmin syötevektoreiden arvoja. Tämänkaltaisen (kaksiulotteisen kartan) algoritmin asymptoottisen suoritusajan ollessa parhaimmillaan $\mathcal{O}(n^2)$ SOM-algoritmi pystyy käsittelemään massiivisia syötteitä visualisoidakseen topologisia malleja (pattern) ja struktuureja huomattavasti ohjatun oppimisen algoritmeja tehokkaammin [8]. [1, 6] On näytetty, että SOM pystyy esimerkiksi piirreirrotuksessa (feature extraction) parempiin suorituksiin kuin tavanomaiset piirreirrotusalgoritmit kuten empiirinen ortogonaalifunktio ja pääkomponenttianalyysi. [9]

3.1 Algoritmi

SOMin muodostamista varten syötteen on oltava vektorimuotoista dataa, jotka kokonaisuutena muodostavat syötematriisin. Oletetaan, että syötematriisi X_{kn} koostuu k vektorista, jotka ovat muotoa $x_k = [x_{k1}, x_{k2}, \dots, x_{kn}]^T \in \mathbb{R}^n$, jossa n on vektorien alkoiden eli ulottuvuuksien määrä. Lisäksi oletetaan, että painovektormatriisi M_{in} koostuu i vektorista, jotka vastaavat muodoltaan syötevektoreita, ja i :nnen neuronin painovektori $m_i = [m_{i1}, m_{i2}, \dots, m_{in}]^T \in \mathbb{R}^n$. [1] Neuroneita alustettaessa painovektoreille annetaan satunnaiset arvot. Neuronien alustamisen jälkeen algoritmin toiminta voidaan jakaa karkeasti seuraaviin vaiheisiin:

1. Valitaan satunnainen vektori syötteestä.
2. Verrataan syötevektoria kaikkien neuronien painovektoreihin siten, että löydetään syötevektoria euklidisesti lähinnä oleva neuroni eli voittajaneuroni.
3. Päivitetään voittajaneuronin ja sen naapurineuroneiden painovektoreita lähemmäs syötevektoria.
4. Palataan vaiheeseen 1, kunnes jokainen syötevektori on käsitelty.

Voittajaneuronia (BMU) valittaessa syötevektorien ja painovektorien ulottuvuuksien yhtäsuuruus on olennaista euklidisen etäisyyden määrittämisen kannalta:

$$d(x_k, m_i) = \sqrt{(x_{k1} - m_{i1})^2 + (x_{k2} - m_{i2})^2 + \dots + (x_{kn} - m_{in})^2}$$

Euklidinen etäisyys d syötevektorin x_k ja painovektorin m_i välillä

Voittajaneuronin valinta voidaan sieventää muotoon

$$\|x - m_c\| = \min_i \|x - m_i\|,$$

jossa m_c on syötevektoriin verrattava painovektori.

SOMEissa euklidinen etäisyys on topologisuuden säilyttämisen perusta. Lisäksi topologisuuden säilyttäminen vaatii, että neuronit edustavat syötettä. Tästä syystä voittajaneuronin painovektoria säädetään lähemmäs syötettä. Toisaalta myös neuroneiden alueelliset osajoukot, naapurustot, ovat olennaisia topologisuuden kannalta, koska voittajaneuronin lähistöllä olevia neuroneita säädetään niin ikään lähemmäs syötettä. Naapuruston neuroneihin tehtävän muutoksen intensiteetti riippuu sekä etäisyydestä voittajaneuroniin että oppimiskertoimesta (learning rate). On havaittu, että alussa leveän naapuruston kaventaminen iteraatioiden edetessä parantaa kartan kokonaisvaltaista jäsentelyä syötteen suhteen. Alussa satunnaisarvoja omaavien painovektoreiden laajamittaisempi siirtely karkeaan järjestykseen edesauttaa myöhemmän vaiheen kapeamman naapurustojoukon hienosäätöä. Naapurustojoukon koko voi suurilukuisella syötteellä lopulta päätyä sisältämään vain yhden neuronin (voittajaneuronin), mutta topologinen järjestys on oltava luotuna tätä ennen. [1]

Neuronin painovektorin päivitysfunktio on yksinkertaisimmillaan muotoa

$$m_i(t+1) = \begin{cases} m_i(t) + \alpha(t)(x(t) - m_i(t)), & \text{if } i \in N_c(t) \\ m_i(t), & \text{if } i \notin N_c(t) \end{cases}$$

jossa t on käynnissä oleva iteraatio, α oppimiskerroin $0 < \alpha(t) < 1$, x käsiteltävä syötevektori ja $N_c t$ valittu naapurusto. Tämänkaltaisen päivitysfunktion naapuruston koko on vakio, ja se päivittää koko valitun naapuruston samalla voimakkuudella kuin itse voittajaneuronin, eikä siten välttämättä ole optimaalisin.

Naapurustofunktion valintaan ei ole yhtä oikeaa tapaa syötteiden mahdollisen monimuotoisuuden takia, mutta tyypillisiä valintoja ovat esimerkiksi voittajaneuroniin keskitetty Gaussin käyrä tai vakioarvolla laskeva funktio. [10] Voittajaneuroniin keskitetty Gaussin käyrän mukaan päivitysfunktio h_{ij}^c

$$h_{ij}^c = \alpha(t) e^{-\left(\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}\right)},$$

jossa α on oppimiskerroin, t käynnissä oleva iteraatio, r_c voittajaneuronin painovektori, r_i i :nneuronin painovektori ja σ naapuruston säde. Tässä tapauksessa σ on Gaussin käyrän keskihajonta kyseisen iteraation kohdalla, joka voidaan määrittää seuraavasti:

$$\sigma(t) = \sigma_0 \cdot e^{(-t/\lambda)},$$

jossa λ on keskihajontaa pienentävä vakio.

Tällöin koko päivitysfunktio on muotoa

$$m_i(t+1) = m_i(t) + h_{ij}^c((x(t) - m_i(t)))$$

Oppimiskertoimen määrittämiseen on olemassa muutamia eri vaihtoehtoja. Useimmiten käytettyjä variantteja ovat lineaariaasien ajan funktio, käänteisen ajan funktio, potenssisarja sekä heuristinen kerroin. Lineaarisen ajan oppimiskerroin on muotoa $\alpha(t) = \frac{1}{t}$.

3.2 Havainnollistus

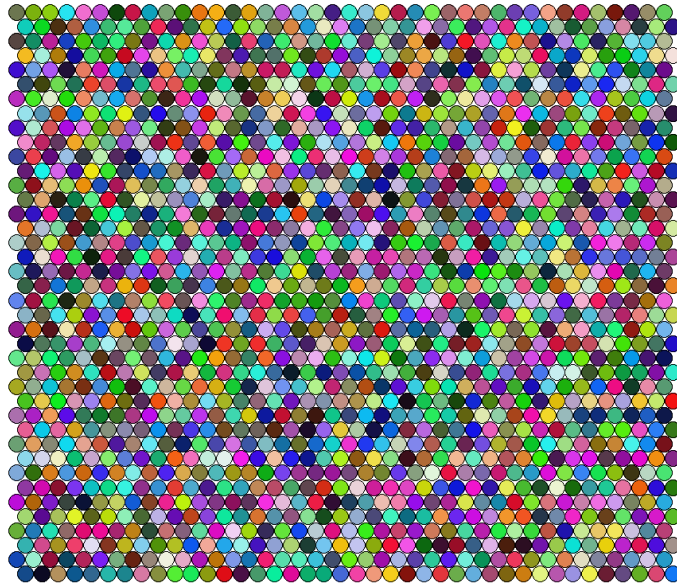
Havainnollistetaan SOMin muodostamista ja kouluttamista R-kielellä RStudio-ohjelmistolla. Oletetaan RGB-värimallin datasta koostuva syötematriisi

$$X \in M_{10000 \times 3}(F), F \in \mathbb{Z} : F \in [0, 255],$$

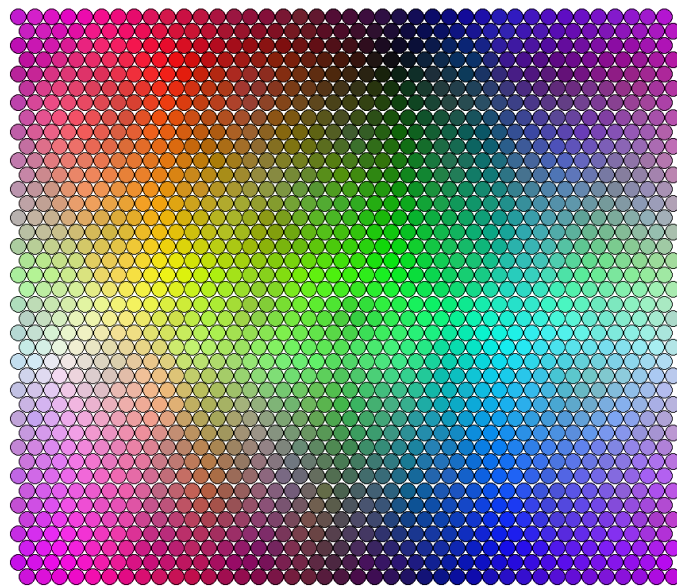
jossa sarakkeet R , G ja B , joten yksi rivi muodostaa yhden värin vektorin. Syöte on luotu valmiiksi tasaisesti jakautuneena ja syötteet ovat keskenään samalla skaalalla, joten normalisointia ei tässä tapauksessa tarvitse suorittaa uudelleen. Huomioitavaa

on, että SOMin muodostamista varten syötteen ei tarvitse olla normalisoitua, mutta siitä on todettu tietyissä tapauksissa olevan hyötyä [1].

Luodaan seuraavaksi kartan neuronimatriisi (kuva 3.1). Yleisesti käytetty määrä neuroneille suhteessa syötteeseen on $M_{max} = 5\sqrt{N}$, jossa N on syötteen koko [11]. Koska valittu syöte ei ole kovin moniulotteista voidaan kartan kokoa nostaa noin kolminkertaiseksi, $M = 16\sqrt{10000}$. Kartan neuroneiden painovektoreille annetaan satunnainen, syötevektorin muotoa vastaava arvo. Kartta itsessään on topologiaaltaan heksagonaalinen ja muodoltaan toroidinen. Naapurustofunktio noudattaa kuplafunktiota. Kuplafunktio on tässä tapauksessa vakioarvo naapuruston säteelle.



Kuva 3.1: Esimerkkitilanne kartasta ennen ensimmäistä iteraatiota, jossa neuroneiden painovektorien arvot on poimittu satunnaisesti syötteestä.



Kuva 3.2: Koulutettu kartta.

Koulutetusta kartasta 3.2 huomataan, että samankaltaiset värit muodostavat ryppäitä, jotka jatkuvat reunojen yli (toroidisuus). Kartalla ei myöskään välttämättä ole täysin samoja värejä kuin alkuperäisessä syötteessä, vaan syötteestä muodos-

tettuja "keskiarvoja". Tärkeimpänä havainnollistuksena on syötteen suhteiden ilmaiseminen. Karttaa voitaisiin jatkaa suorittamalla esimerkiksi k :n keskiarvon klusterointi.

4 Muunnelmat ja laajennokset

Vaikka SOMia voidaan hyödyntää lukuisissa erilaisissa tehtävissä, sen toiminta ei kuitenkaan ole täysin ongelmaton. Eräänä perustavanlaatuisena ongelmana voidaan pitää esimerkiksi sitä, että naapuruston koko on määritettävä etukäteen, eikä muodostettavan kartan naapurustofunktion valintaan ole olemassa optimaalista, syötteen tyyppin huomioivaa ratkaisukaavaa. Karttaa on siksi testattava useilla eri naapurustofunktioilla parhaan tuloksen löytämiseksi. Toisena ongelmana voidaan pitää kartan kokoa, joka on myös määriteltävä etukäteen. Koon valintaan on olemassa vakiintuneita käytäntöjä, mutta varsinaista täydellistä valintakaavaa ei ole. Lisäksi on huomioitava, että liian pieni kartta ei kykene edustamaan syötettä riittävän tarkasti. Muina ongelmina voidaan nähdä BMU:n etsinnän lineaarinen aikavaativuus sekä neuronien tai neuroniryppäiden konvergoituminen epäsuotuisasti tilanteessa, jossa syötedata on jakautunut epätasaisesti koko syöteavaruuteen. Tällöin jotkin neuronit saattavat jäädä naapurustofunktion ulkopuolelle "hylätyiksi" ja siten antaa vääränlaisen kuvan syötteestä. [12] Näiden ongelmien vuoksi SOMista on kehitetty lukuisia muunnelmia ja laajennoksia. Varianttien kehittämisen tarkoituksena on voinut olla myös tarve kehittää tietyn tutkimuksen dataan paremmin sopiva malli.

Tyypillisesti variantit laajentavat tai muokkaavat alkuperäisen SOMin algoritmia, säilyttäen kuitenkin itseorganisoituvuuden peruseriaatteen. Variantiksi voidaan myös lukea malli, joka noudattaa SOMin algoritmia, mutta visualisoi muodos-

tettua karttaa halutulla tavalla. Monet uusimmat variantit toteuttavat molempia ja laajentavat jotakin toista varianttia. Seuraavassa perehdytään tarkemmin muutamaamaan yleisesti käytettyyn varianttiin, niiden eroavaisuuksiin alkuperäisestä SOMista sekä niiden kehittämisen syihin. Tutkielmaan valitut variantit voidaan jakaa kahteen tyyppiin: GSOM ja ESOM on kehitetty yleisellä tasolla ratkaisemaan SOMin ongelmakohtia muokkaamalla SOMin algoritmia, kun taas FlowSOM ja NOASSOM on kehitetty kohdennetusti tietyn alan tutkimusta varten jatkojalostamalla muodostettua karttaa.

4.1 Growing SOM, GSOM

Growing self-organizing map (GSOM) eli kasvava itseorganisoidun kartta on variaatio, joka nimensä mukaisesti kasvaa iteraatioiden edetessä. Yhtäältä variaation tarkoitus on poistaa tarve valita kartan koko ennen kartan muodostamista, toisaalta antaa tutkijalle parempi lähtökohta klustereiden tarkasteluun. GSOMin muodostus aloitetaan minimaalisella, yleensä 2×2 ruudukolla, jossa neuroneita lisätään reu-
nimmaisten neuronien naapureiksi [13]. Toisin kuin SOM, GSOM tukee ainoastaan suorakulmaista muotoa, jonka on todettu olevan heikompi kuvastamaan syötettä verrattuna heksagonaaliseen muotoon. Vuonna 2003 julkaistiin versio GSOMista, joka tukee myös heksagonaalista muotoa. [14]

Itseorganisoiduvien karttojen avulla luotaessa ja tarkastellessa klustereita on hyödyllistä saada jo kartan muodostusvaiheessa tietoa merkittävimmistä klustereista, sillä siten kyetään keskittämään neuroneita merkittävimpiin klustereihin, mikä edesauttaa tarkempien klustereiden muodostamista. Samalla voidaan mahdollisesti kyetä tekemään päätöksiä klustereista, jotka eivät välttämättä ole tarkastelun kannalta merkittäviä. GSOMin kasvumekanismi mahdollistaa kartan kasvattamisen tarkemmin niille alueille, jotka ovat tarkastelulle enemmän oleellisia. Tämänkaltaisen kasvu voidaan saavuttaa painottamalla kasvua esimerkiksi syötteen jonkin tietyn attribu-

tin eli ulottuvuuden mukaan. Kasvu suoritetaan käyttäen *kasvukerrointa* (spread factor). Kasvukertoimen lisäksi GSOM tarvitsee parametrit *kasvun kynnyisarvo* GT (*growth threshold*) sekä *virhearvo* E (*error value, error counter*).

GSOMin muodostamisessa on kolme selkeää vaihetta: alustus-, kasvu- sekä tasoitusvaihe. Itseorganisoituvuus tapahtuu kasvu- ja tasoitusvaiheissa, joskin naapurusto- ja päivitysfunktioiden parametreja muokataan hieman. Seuraavassa kuvataan GSOM-algoritmia pääpiirteittäin.

1. Alustusvaihe

1.1 Alustetaan ensimmäisten neuroneiden painovektorit satunnaisilla arvoilla.

1.2 Lasketaan kasvun kynnyisarvo GT kaavalla

$$GT = -D \times \ln(SF),$$

jossa D on syötteen ulottuvuuksien määrä ja SF kasvukerroin. Kasvukerroin saa arvon nolasta yhteen, jossa 0 edustaa minimaalista ja 1 maksimaalista kasvua. Oletettavaa on, että kasvukertoimen arvo iteraatioiden alussa on pieni. Kerrointa voidaan kasvatettaa asteittain iteraatioiden edetessä niillä alueilla, joiden tahdotaan kasvavan voimakkaammin. [13]

2. Kasvuvaihe

2.1 Valitaan syötevektori, etsitään BMU ja päivitetään BMU:n sekä sen naapuruston painovektorit kuten SOMissa. Huomioitavaa on, että GSOMin lähtökohtainen naapuruston koko on yleensä pienempi, sillä neuroneita on alussa vähän.

2.2 BMU:n virhearvoa kasvatetaan kaavalla

$$E(t + 1) = E(t) + \|x(t) - m_c\|,$$

jossa x syötevektori ja m_c BMU:n painovektori.

- 2.3** Jos BMU on kartan reunaneuroni ja jos BMU:n $E > GT$, kasvatetaan karttaa lisäämällä neuroni kaikkiin BMU:n tyhjiin naapuripaikkoihin. Uusien neuroneiden painovektorit saavat arvon

$$w_{uusi} = 2w_{bmu} - w_o,$$

jossa w_o uutta neuronina topologisesti vastapäätä olevan neuronin painovektori. Jos vastapäistä neuronina ei ole olemassa painovektori on

$$w_{uusi} = w_{bmu} + w_a - w_b,$$

jossa w_a ja w_b uutta neuronina lähinnä olevat kaksi neuronina, jotka eivät kuitenkaan ole BMU. Jos $E > GT$, mutta BMU ei ole reunaneuroni, E propagoidaan BMU:n naapureille.

- 2.4** Toistetaan vaiheet 2.1-2.3 kunnes koko syöte on käyty läpi.

3. Tasoitusvaihe

- 3.1** Alustetaan oppimiskerroin kasvuvaiheeseen verrattuna pienemmällä arvolla ja valitaan naapurustoksi pienempi, yleensä neuronin välitön ympäristö.

- 3.2** Iteroidaan syöte läpi kuten kasvuvaiheessa, mutta ilman kasvua.

4.2 Evolving SOM, ESOM

Evolving self-organizing map on SOMin variaatio, jonka tarkoituksena on vähentää oppimiseen tarvittavaa aikaa ja maksimoida käytetyn ajan tehokkuus. ESOMin esitteli Da Deng ja Nikola Kasabov vuonna 2003. [15] ESOM kehitettiin toimimaan

adaptiivisesti syötteeseen reaaliaikaisesti lisättävään dataan, jonka kanssa SOM ei juurikaan kykene toimimaan luotettavasti. Muina hyötyinä SOMiin verrattuna voidaan pitää vähäisempää geometrista rajoittuneisuutta, kompaktimpaa neuroniverkkoa sekä syötettä tarkemmin kuvaavaa mallinnusta.[16]

ESOMin verkkotopologia eroaa SOMista huomattavasti. Kartta alustetaan ilman neuroneita ja siten ilman valmiiksi valittua topologiaa, eivätkä myöhemmin luotavat neuronit ole sidottuina karttaan tietyille paikoille. Neuronit yhdistävän verkon puuttuessa neuroneiden väliset naapurisuhteet käsitellään etäisyyksinä. Oppimisen aikana kartta päivittää itseään luoden tarvittaessa uusia neuroneita. Etäisyydet määrittävät kahden neuronin välisen naapurisuhteen voimakkuuden. Heikoimmat suhteet on mahdollista karsia pois, jolloin kartta voi repeytyä useampaan osaan ja muodostaa ryppäitä sekä osoittaa merkittävästi poikkeavia havaintoja. Neuroneiden painovektorit ovat samankaltaisia kuin SOMissa, vastaten ulottuvuudellisesti syötevektoria. [15, 16]

ESOMin neuroneille annetaan luontivaiheessa arvoja suoraan syöttestä satunnaisten arvojen sijaan. Tästä seuraa se, että kartalle annettaessa uutta syötettä oppiminen ei ala alusta, vaan kartalla on muisti edeltävästä syöttestä. Oppiminen jatkuu sovittamalla vanhaa olemassaolevaa muistia muuttuvaan syötteeseen. Oppiminen ESOMissa on paikallisempaa verrattuna SOMiin, jolloin vältetään SOMin rajoitteellisuutta esimerkiksi niissä tilanteissa, missä jotkin neuronit ovat "lukituina" pysymään keskellä verkon topologian takia. Vastaavasti ESOMissa merkityksettömien neuroneiden määrä on vähäisempi. ESOMin suorituskustannus pienenee SOMiin verrattuna neuroneiden alustuessa valmiiksi syötteiden kaltaisiksi. [16]

4.3 FlowSOM

FlowSOM on erityisesti virtaus- ja massasytometrian visualisointiin ja klusterointiin kehitetty SOM-variantti, joka on julkaistu vuonna 2015. FlowSOM muodostaa tavan-

omaisen SOMin algoritmin mukaan kartan, jota jatketaan jatkoanalysointia varten muodostamalla pieni virittävä puu (minimal spanning tree) sekä suorittamalla metaklusterointia. FlowSOMin muistuttaa toista sytometriassa käytettyä algoritmia, SPADEa (Spanning-tree Progression Analysis of Density-normalized Events), joka suorittaa hierarkkista klusterointia. FlowSOMin on raportoitu saavuttaneen useista verratuista malleista suoritustehokkaimman tuloksen syötteellä, jossa populaatioita useita tai jossa populaatiot ovat harvalukuisia. FlowSOMin heikkoutena voidaan pitää tarvetta määrittää etukäteen optimaalinen määrä ennustettavia klustereita, etenkin niissä tilanteissa joissa tarkoituksena on kartoittaa harvalukuisia populaatioita. [17, 18, 19]

4.4 NOASSOM

NOASSOM (Nonlinear Orthogonal Adaptive-Subspace Self-Organizing Map) on laajennos, joka on kehitetty tunnistamaan videomateriaalista esiintymiä ja liikettä määrittääkseen videon olennaisimpia alueita. NOASSOM jatkaa Kohosen vuonna 1995 esittelemää ASSOM-laajennosta (Adaptive-Subspace Self-Organizing Map), joka on SOMin modulaarinen muunnos. Sen moduulit oppivat tunnistamaan malleja (pattern) syötteestä, jossa tapahtuu muutoksia. Tämän vuoksi ASSOM kykenee käsittelemään virtaustyyppistä dataa, kuten esimerkiksi videomateriaalin kuvaruudut (frame). Moduulit muodostavat aliavaruuskerroksen syöte- ja ulostulokerroksen väliin. [20] NOASSOM lisää vielä yhden, non-lineaarisen ortogonaalisen kartan, kerroksen syöte- ja aliavaruuskerroksen väliin. NOASSOM hyödyntää myös toisia algoritmeja, kuten tiheän liikeradan ja Fisher-vektorin algoritmeja. Laajennos esiteltiin vuonna 2018. Laajennoksen tarkoituksena on ASSOMiin verrattuna yhtäältä toimia paremmin epälineaarisen datan kanssa, jota tämänkaltaisen käytännön data yleensä on, ja toisaalta pysyä alkuperäisen SOMin tavoin ohjaamattoman oppimisen mallina. ASSOM luottaa merkittyyyn dataan, joten sitä ei voida pitää ohjaamattomana. [21]

4.5 Muut laajennokset

SOMin kehitys julkaisua seuranneena vuosikymmenenä keskittyi enimmäkseen teoreettisten ja matemaattisten ominaisuuksien ja rajoitteiden tutkimiseen, sekä algoritmin optimointiin. 1990-luvulla SOM yleistyi klusterointimenetelmänä, jolloin ilmaantui ensimmäisiä SOMin algoritmia muokkaavia variantteja, kuten tässä luvussa esitetty GSOM. Muita saman aikakauden variantteja ovat esimerkiksi fuzzy-logiikkaa SOMiin yhdistävä fuzzy self organizing map (FSOM) ja Bayesian self-organizing map (BSOM), jossa neuronit edustavat Gaussin jakaumia [22, 23]. Growing neural gas (GNG) on saanut innoitusta SOMilta, mutta sitä ei varsinaisesti pidetä SOMin laajennoksena [24]. 2000-luvulla julkaistuja variantteja tässä luvussa esitellyn ESOMin lisäksi ovat muun muassa robust SOM (RSOM), growing hierarchical SOM (GHSOM) ja incremental SOM (ISOM) [25, 26, 27]. Viimeisen vuosikymmenen aikana SOMista on kehitetty variantteja, joihin on yhdistelty esimerkiksi syväoppimista, kuten deep learning SOM (DLSOM), ja 3D-mallinnusta kuten cube Kohonen SOM (CKSOM) [28, 29]. Kehityksen kulusta voidaan huomata, että SOM ei varsinaisesti edusta teknologisesti uraauurtavaa kehitystä, vaan on etenkin 2000-luvulle tultaessa siirtynyt hyödyntämään ja integroimaan itseensä muita koneoppimismalleja, antaen vaihtoehtoisia tapoja eri tyyppisten ongelmien analysointiin. Tästä huomataan SOMin poikkeuksellinen potentiaali muuntautua ja laajentua toimimaan erilaisissa tehtävissä eri tutkimusalojen välillä.

5 Yhteenveto

Tässä tutkielmassa tarkasteltiin itseorganisoituvan kartan algoritmia ja muodostamista, sekä neljää erilaista SOMin laajennosta ja niiden eroavaisuuksia. Itseorganisoituva kartta on biologisesta neuroverkosta innoituksensa saanut koneoppimismalli, jota käytetään laajasti eri tutkimusaloilla moniulotteisen datan visualisointiin ja tulkintaan. SOM muodostaa syötteen alkioita edustavan kartan, joka säilyttää syötteessä esiintyvät topologiset vaihtelut.

SOMin etuina voidaan nähdä muuntautuvuuskyky, luotettavuus ja tehokkuus. SOMin käytöstä ja vertailusta toisiin algoritmeihin eri alan tehtävissä löytyy runsaasti aineistoa, mikä tekee siitä kattavasti tutkitun vaihtoehdon toisille koneoppimismalleille. Optimaalisten tulosten saavuttamiseksi SOMin parametrit on kuitenkin hienosäädettävä, joka vaatii syötteen kontekstuaalista ymmärtämistä, ja on usein saavutettavissa yritys-erehdys -menetelmällä. Teknologian ja laskentatehon kehityessä analysoitavan datan määrä ja kompleksisuus on kasvanut, jolloin alkuperäisestä SOMista on tarvinnut kehittää tehokkaampia ja dynaamisempia variaatioita. Tutkielmassa käsiteltiin variantteja, joista on saatavilla tutkimustietoa ja sen myötä vertailutuloksia toisiin malleihin. Näistä varianteista luvussa 4 esiteltyt GSOM ja ESOM ovat SOMin algoritmin muunnoksia ja yleisellä tasolla käytännöllisiä. Ne pyrkivät ratkaisemaan SOMin fundamentaalisia ongelmia, kuten tarvetta valita kartan koko etukäteen ja toimimattomuutta muuttuvan syötteen kanssa. Luvussa 4 esiteltyt FlowSOM ja NOASSOM puolestaan ovat spesifimpiä, tietyn tyyppiseen tutki-

mukseen tarkoitettuja malleja. FlowSOM ei muuta SOMin algoritmiä, vaan jatkaa saadun kartan työstämistä. NOASSOM on ASSOM-muunnelman muunnelmä, joka muokkaa edeltäjänsä algoritmia paremmin toimivaksi NOASSOMille tyypillisen syötteen kanssa.

SOMin kehitykselle on tyypillistä, että algoritmia muutetaan käsiteltävälle dataalle sopivammaksi. Tämä voidaan nähdä sekä vahvuutena että heikkoutena: kyseisen tutkimuksen syöttelelle voidaan saavuttaa mahdollisimman oikeellisia tuloksia antava ja optimaalisesti toimiva algoritmi, toisaalta parhaan mahdollisen algoritmin luominen on työlästä ja sen käyttökohde saattaa olla hyvin marginaalinen. SOMien käyttöön ja kehitykseen on saatavilla erilaisia paketteja R ja Python -kielille, joiden myötä SOMin käyttö ja kehitystyö on entistä saavutettavampaa.

Lähdeluettelo

- [1] T. Kohonen, ”The self-organizing map”, vol. 78, nro 9, s. 1464–1480, 1990. DOI: 10.1109/5.58325.
- [2] D. Miljković, ”Brief review of self-organizing maps”, teoksessa *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, IEEE, 2017, s. 1061–1066. DOI: 10.23919/MIPRO.2017.7973581.
- [3] T. Kohonen, S. Kaski, K. Lagus et al., ”Self organization of a massive text document collection”, teoksessa *Kohonen Maps*, E. Oja ja S. Kaski, toim., Amsterdam: Elsevier Science B.V., 1999, s. 171–182. DOI: 10.1016/B978-044450270-4/50013-9.
- [4] A. Jain, J. Mao ja K. Mohiuddin, ”Artificial neural networks: a tutorial”, *Computer*, vol. 29, nro 3, s. 31–44, 1996. DOI: 10.1109/2.485891.
- [5] G. Deboeck, ”Financial Applications of Self-Organizing Maps”, *Neural Network World*, vol. 8, toukokuu 2000.
- [6] J. Buhmann ja H. Kuhnel, ”Unsupervised and supervised data clustering with competitive neural networks”, teoksessa *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, vol. 4, IEEE, 1992, s. 796–801. DOI: 10.1109/IJCNN.1992.227220.

-
- [7] M. B. Gorzałczany ja F. Rudziński, "Evolution of SOMs' Structure and Learning Algorithm: From Visualization of High-Dimensional Data to Clustering of Complex Data", *Algorithms*, vol. 13, nro 5, 2020. DOI: 10.3390/a13050109.
- [8] D. Roussinov ja H.-c. Chen, "A Scalable Self-Organizing Map Algorithm for Textual Classification: A Neural Network Approach to Thesaurus Generation", *CC-AI*, vol. 15, lokakuu 1999.
- [9] Y. Liu, R. H. Weisberg ja C. N. K. Mooers, "Performance evaluation of the self-organizing map for feature extraction", *Journal of Geophysical Research: Oceans*, vol. 111, nro C5, 2006. DOI: doi.org/10.1029/2005JC003117.
- [10] M. P. J. Tian M. H. Azarian, "Anomaly Detection Using Self-Organizing Maps-Based K-Nearest Neighbor Algorithm", *PHM Society European Conference*, vol. 2, heinäkuu 2014. DOI: 10.36001/phme.2014.v2i1.1554.
- [11] J. Vesanto ja E. Alhoniemi, "Clustering of the self-organizing map", *IEEE Transactions on Neural Networks*, vol. 11, nro 3, s. 586–600, 2000. DOI: 10.1109/72.846731.
- [12] C. A. Astudillo, "Self organizing maps constrained by data structures", tohtorinväitöskirja, 1125 Colonel-By Drive Ottawa, Ont. K1S 5B6 Canada, 2011, ISBN: 9780494815441.
- [13] D. Alahakoon, S. Halgamuge ja B. Srinivasan, "Dynamic self-organizing maps with controlled growth for knowledge discovery", *IEEE Transactions on Neural Networks*, vol. 11, nro 3, s. 601–614, 2000. DOI: 10.1109/72.846732.
- [14] A. L. Hsu, S.-L. Tang ja S. K. Halgamuge, "An unsupervised hierarchical dynamic self-organizing approach to cancer class discovery and marker gene identification in microarray data", *Bioinformatics*, vol. 19, nro 16, s. 2131–2140, marraskuu 2003. DOI: 10.1093/bioinformatics/btg296.

- [15] D. Deng ja N. Kasabov, "ESOM: an algorithm to evolve self-organizing maps from online data streams", teoksessa *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, vol. 6, IEEE, 2000, 3–8 vol.6. DOI: 10.1109/IJCNN.2000.859364.
- [16] X. Qiang, G. Cheng ja Z. Li, "A survey of some classic self-organizing maps with incremental learning", teoksessa *2010 2nd International Conference on Signal Processing Systems*, vol. 1, IEEE, 2010, s. V1-804-V1-809. DOI: 10.1109/ICSPS.2010.5555247.
- [17] S. Van Gassen, B. Callebaut, M. J. Van Helden et al., "FlowSOM: Using self-organizing maps for visualization and interpretation of cytometry data", *Cytometry Part A*, vol. 87, nro 7, s. 636–645, 2015. DOI: doi.org/10.1002/cyto.a.22625.
- [18] S. Montante ja R. R. Brinkman, "Flow cytometry data analysis: Recent tools and algorithms", *International Journal of Laboratory Hematology*, vol. 41, nro S1, s. 56–62, 2019. DOI: <https://doi.org/10.1111/ijlh.13016>.
- [19] L. M. Weber ja M. D. Robinson, "Comparison of clustering methods for high-dimensional single-cell flow and mass cytometry data", *Cytometry Part A*, vol. 89, nro 12, s. 1084–1096, 2016. DOI: <https://doi.org/10.1002/cyto.a.23030>.
- [20] T. Kohonen, S. Kaski ja H. Lappalainen, "Self-Organized Formation of Various Invariant-Feature Filters in the Adaptive-Subspace SOM", *Neural Computation*, vol. 9, nro 6, s. 1321–1344, 1997. DOI: 10.1162/neco.1997.9.6.1321.
- [21] Y. Du, C. Yuan, B. Li et al., "Hierarchical Nonlinear Orthogonal Adaptive-Subspace Self-Organizing Map Based Feature Extraction for Human Action

- Recognition”, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, nro 1, s. 6805–6812, 2018. DOI: 10.1609/aaai.v32i1.12248.
- [22] P. Vuorimaa, ”Fuzzy self-organizing map”, *Fuzzy Sets and Systems*, vol. 66, nro 2, s. 223–231, 1994. DOI: [https://doi.org/10.1016/0165-0114\(94\)90312-3](https://doi.org/10.1016/0165-0114(94)90312-3).
- [23] H. Yin ja N. Allinson, ”Bayesian learning for self-organising maps”, *Electronics Letters*, vol. 33, 304–305(1), 4 1997. DOI: 10.1049/e1:19970196.
- [24] B. Fritzke, ”A Growing Neural Gas Network Learns Topologies”, teoksessa *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky ja T. Leen, toim., vol. 7, MIT Press, 1994, s. 625–632. url: https://proceedings.neurips.cc/paper_files/paper/1994/file/d56b9fc4b0f1be8871f5e1c40c0067e7-Paper.pdf.
- [25] H. Allende, S. Moreno, C. Rogel ja R. Salas, ”Robust Self-organizing Maps”, teoksessa *Progress in Pattern Recognition, Image Analysis and Applications*, A. Sanfeliu, J. F. Martinez Trinidad ja J. A. Carrasco Ochoa, toim., Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, s. 179–186. DOI: 10.1007/978-3-540-30463-0_22.
- [26] A. Rauber, D. Merkl ja M. Dittenbach, ”The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data”, *IEEE Transactions on Neural Networks*, vol. 13, nro 6, s. 1331–1341, 2002. DOI: 10.1109/TNN.2002.804221.
- [27] A. P. Paplinski, ”Incremental Self-Organizing Map (iSOM) in Categorization of Visual Objects”, teoksessa *Neural Information Processing*, T. Huang, Z. Zeng, C. Li ja C. S. Leung, toim., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, s. 125–132. DOI: 10.1007/978-3-642-34481-7_16.

-
- [28] W. Lee, S. Hasan, S. Shamsuddin ja N. Lopes, ”Deep learning SOM library for surface reconstruction”, *International Journal of Advances in Soft Computing and its Applications*, vol. 9, s. 1–16, 2017, ISSN: 2074-8523.
- [29] S. P. Lim ja H. Haron, ”Cube Kohonen Self-Organizing Map (CKSOM) Model With New Equations in Organizing Unstructured Data”, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, nro 9, s. 1414–1424, 2013. DOI: 10.1109/TNNLS.2013.2259259.