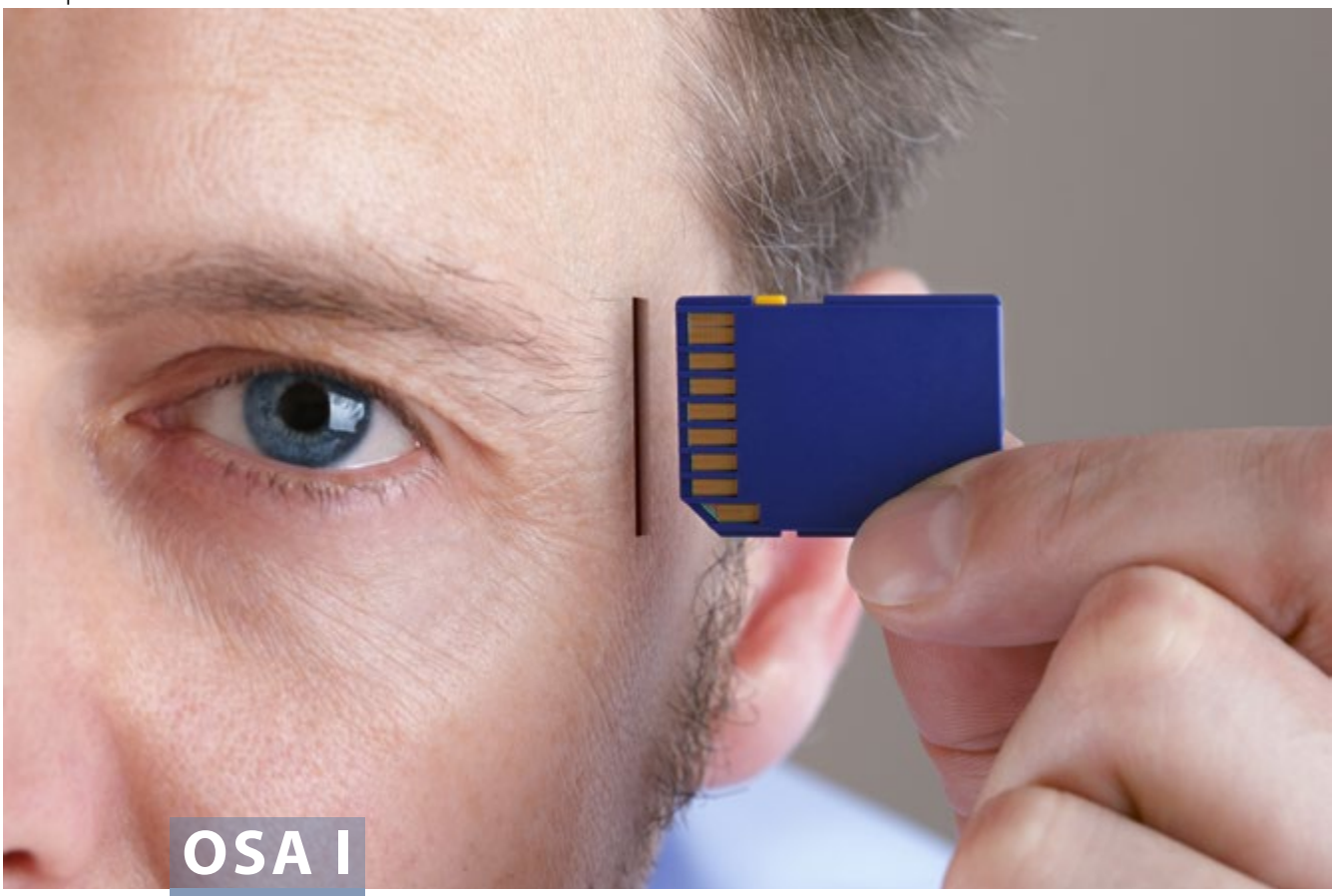
iStock | BrianAJackson



OSA I

TEKOÄLYN KANSSA ASUNTOKAUPOILLA

Ari Laitala

TEKOÄLYÄ ON SOVELLETTU kiinteistömarkkinoiden mallintamiseen ja kiinteistöjen arvonmääritykseen ainakin jo 1990-luvun alkupuolelta lähtien. Suhteellisen varhaisia tutkielmia aihepiiristä ovat julkaisseet mm. Do&Grudnitski 1992, Worzala et al. 1995 ja Rossini 1997. Nämä artikkelit löytyvät myös googlaamalla, joten uteliaat voivat niihin helposti tutustua. Vaikka aihetta on akateemisissa piireissä pyöritelty jo ainakin neljännesvuosisadan ajan, on käytännön sovelluksista kiinteistömarkkinoilla edelleen varsin vähän julkista tietoa. Olisi kuitenkin pieni ihme – jollei vähän isompikin – jos menestyneimmät kiinteistösijoittajat eivät olisi käyttäneet näitä sovelluksia jo pitkään. Osaavasti käytettynä tekoälysovellukset näyttäisivät tarjoavan etulyöntiaseman, esim. hinnanmääritystehtävissä tai vaikkapa kysyntää/ tarjontaa ennustettaessa. Tässä valossa jopa varsin tehokkaiden markkinoiden lyöminen tuntuisi mahdolliselta – ainakin niin kauan, kun näitä sovelluksia ei laajemmin käytetä.

Yksi tekoälysovellusten etu on niiden suunnaton oppimisnopeus. Lähdimme ottamaan selvää, kuinka nopeasti tekoäly pääsee jyvälle pääkaupunkiseudun asuntojen hintojen muodostuksesta.

Esimakua tulevasta on saatu myös *Maankäytön* sivuilla jo vuonna 1997 silloisen TKK:n tutkijan **Markus Törmän** mainiossa artikkelissa ”Neuraaliverkot ja niiden käyttö kuvien analysoinnissa” (*Maankäyttö* 1/97). Törmän artikkelissa perehdytään keinotekoisten neuraaliverkkojen käyttöön (satelliitti)kuvien tulkinnassa, mikä onkin pitkään ollut yksi isoimmista tekoälyn sovellusalueista. Nykyään tekoälyä käytetään varsin laajalti myös puheen, liikkeen ja äänen tunnistamiseen. Sinänsä neuroverkkolaskennan ja algoritmien perusteet eivät ole parissa vuosikymmenessä muuttuneet juuri lainkaan. Törmänkin artikkeli on edelleen hyvin validi ja suurin muutos lienee siinä, että nykyään on muodikkaampaa puhua neuroverkoista kuin neuraaliverkoista.

Suurin kehitysaskel näyttääkin tapahtuneen rautapuolella. Nykyään jo halpisläppärikin selviää melko vaativista laskentatehtävistä siedettävässä ajassa.

MIKÄ ON TEKOÄLY?

Tekoälyuutiset alkavat olla jo melkein päivittäistä uutisvirtaa. Esimerkiksi *Kauppalehti* uutisoi 31.3.2017 tekoälyn vievän osan

varsinkin nuorempien juristien työstä. Eivätkä tekoälysovellukset näytä tyytyvän vain työmaiden valloitukseen. Ylen uutinen 2.4.2017 kertoo data-analyytikko **Johannes Ärjen** kehittämästä vedonlyöntirobotista, joka tienaa pääosin yöaikaan pelailemalla. Ko. jutusta ei saa kovin tarkkaa kuvaa siitä, onko Ärjen itsensä kehittämä ohjelmistorobotti etukäteen määriteltyihin algoritmeihin perustuva vaiko oppimiseen kykenevä ohjelmisto. Kyseessä vaikuttaisi olevan keinotekoinen neuroverkko, joka näin ollen kykenee parantamaan suoritustaan käsitellyn datamäärän kasvaessa.

Mutta mistä oikeastaan puhumme silloin, kun puhumme tekoälystä? Tekoäly on (tietysti) varsin vaikeasti hahmottuva aihepiiri. Käsitteenmäärittelyn juuriongelmaksi näyttäisi paljastuvan osakäsite äly. Mitä äly/älykkyys ylipäänsä on? Turvaudumme tässä kohtaa Wikipediaan, joka kelpaa lähteeksi tämän artikkelin tarkoitusperiin. Wikipedian määritelmässä lähtökohtana on se, että tekoäly (artificial intelligence / AI) on tietokone tai tietokoneohjelma, joka kykenee "älykkäiksi laskettaviin toimintoihin". Varsin lähellä kehäpäätelmää siis ollaan. Asia valaistuu kuitenkin hiukan lisää, kun kerrotaan, että tekoäly voidaan jakaa vahvaan ja heikkoon tekoälyyn. Vahvalla tekoälyllä näytetään yleisesti tarkoitettavan ihmismäistä toiminnallisuutta. Heikko tekoäly puolestaan viittaa kapean osa-alueen toiminnallisuuteen.

Tämän artikkelin esimerkkitapauksena on keinotekoinen neuroverkko (Artificial Neural Network / ANN), jota sovelletaan asuntojen hintojen mallinnukseen. Kyseessä on siis erittäin kapean osa-alueen tekoäly ja siten myös hyvin heikko tekoäly, vaikka se sinällään suorittaakin vaikuttavan määrän laskutoimituksia osana oppimistaan.

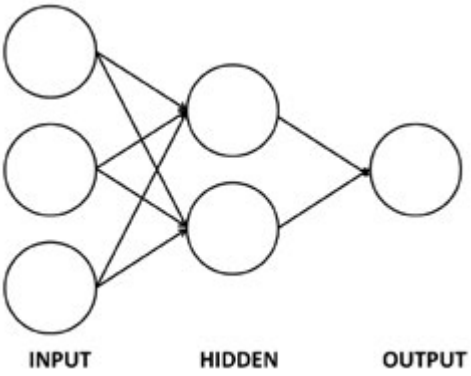
Neuroverkko on pohjimmiltaan kuitenkin hyvin monikäyttöinen työkalu, joka taipuu vahvojenkin tekoälysovellusten työrukkaseksi. Oleellista on se, millaisiin toimintoihin neuroverkko datan avulla opetetaan.

MITEN VERKKO PUNOTAAN?

Tyypillisesti neuroverkot ovat mallia FF eli Feed Forward -verkkoja ja suosituin oppimisalgoritmi näyttäisi olevan backpropagation. Suomenkieliset termien vastineet tuntuvat kankeilta. Ainakin termejä *eteenpäin syöttävä* (FF) ja *takaisinlevittävä* (backpropagation) näkee joissakin graduissa, mutta enemmän kuitenkin alkuperäisiä englanninkielisiä termejä. Myös tässä artikkelissa käytetään voittopuolisesti englanninkielisiä nimityksiä.

Lähdetään seuraavaksi luomaan neuroverkkoa. Olkoon se mallia FF backpropagation -algoritmilla. Verkon yleinen rakenne selviää kuviosta 1. Kuvan verkossa nuolet kuvaavat varsinaisen laskennan suuntaa. Takaisinkytkentöjä ei ole eli varsinainen laskenta-algoritmi laskee siis vain eteenpäin: tästä nimitys Feed Forward. Englanninkielisten mielestä tällainen verkko on nimeltään MultiLayerPerceptron (MLP). Sana perceptron viittaa neuronin tyyppiin. Alun perin perceptron on tarkoittanut neuronia, jonka output on binäärinen eli 0 tai 1. Sittemmin perceptron-termin merkitys näyttää yleistyneen koskemaan muunkin tyyppisiä neuroneita. Neuroneista käytetään myös nimitystä solmu (node).

Vasemmalla oleva Input-kerros lukee tyypillisesti muuttujien arvot (rivi kerrallaan). Tämän artikkelin esimerkissä on kyse asuntokauppadatasta, joka määritellään laskettavaksi niin, että kukin vertailukauppa luetaan kauppa kerrallaan verkkoon input-neuroneilla. Tässä kohtaa huomaamme ensimmäisen samankaltaisuuden suhteessa regressioyhtälöihin. Neuroverk-

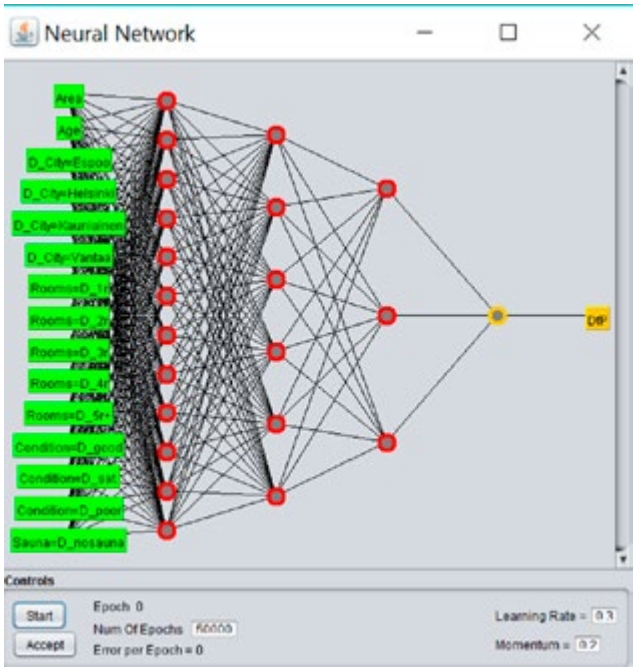


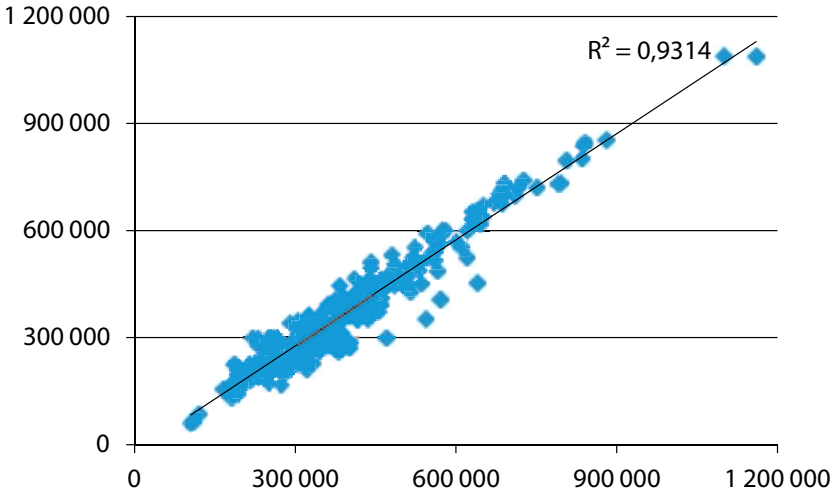
KUVIO 1.

ANN-verkon perusrakenne. Verkon kolme kerrosta muodostuvat (keinotekoisista) neuroneista, jotka on tyypillisesti järjestetty riveihin. Tämän verkon voisi nimetä 3,2,1-verkoksi kerroksissa olevien neuronien lukumäärän mukaisesti (luetaan vasemmalta oikealle).

KUVIO 2.

Käynnissä oleva verkon laskenta. Kuvasta on luettavissa, että aineisto kokonaisuudessaan on ehditty käydä läpi kuvakaappauksen kohdalla 31 176 kertaa (Epoch). Laskenta on säädetty päättymään 100 000 epookin jälkeen.





KUVIO 3.

Ylioppimisen tulos. Selitysaste on noussut ”luonnottoman” korkeaksi, kun verkon on ”annettu” mallintaa myös hinnanmuodostukseen liittyvää satunnaisuutta. X-akselilla on maksettu hinta ja y-akselilla mallinnettu hinta. Muutamia isojakin poikkeamia kuitenkin edelleen havaitaan.

kokin tarvitsee dataa jota laskea. Datan esimerkkirivi voisi olla vaikkapa muotoa:

(ikä, pinta-ala, Sauna_dummy) = (25, 48, 0)

Kyseisen vertailukaupan ikä olisi siis 25 vuotta, pinta-ala 48 neliötä ja kohteessa ei olisi huoneistosaunaa. Jos olisi, pitäisi Sauna_dummyn arvon olla 1 eikä 0. Input-kerroksen ylin neuroni lukisi siis muuttujan iän arvon, toinen neuroni muuttujan pinta-ala arvon ja alin neuroni Sauna-dummyn arvon. Varsinainen laskenta suoritetaan verkon piilokerroksessa, jonne muuttujien arvot siirretään (monistetaan) neuronien välisiä yhteyksiä pitkin. Näillä yhteyksillä on erilainen painoarvo, joka vaikuttaa piilokerroksen laskentaan. Tätä on tarkemmin selitetty jutun yhteydessä olevassa nettiekstrassa *Miten verkko oppii?*

Jo ennen muuttujien luentaa jatkuvien muuttujien arvot valmistellaan laskentaan sopivaksi suorittamalla muuttujien skaalaus. Tyypillisiä ratkaisuja skaalauksessa ovat muuttujien standardointi tai normalisointi. Standardoinnissa jatkuvat muuttujat viedään standardoidulle normaalijakaumalle N~(0, 1) eli kunkin muuttujan arvot lasketaan uusiksi siten, että muuttujan keskiarvoksi saadaan 0 ja keskihajonnaksi 1. Muuttujien normalisoinnissa muuttujien arvot lasketaan uusiksi muuttujien min ja max arvoihin perustuen siten, että muuttujien arvot asettuvat välille [–1, 1] tai joskus [0, 1]. Muuttujien skaalauksen tarkoituksena on verkon laskennan nopeuttaminen.

SITTEN LASKETAAN

Verkon laskentahaaste valitaan vaativimmasta päästä: omakotikiinteistön hintamalli ja vieläpä suhteellisen niukoilla tiedoilla. Vertailukauppa-aineiston hankinnassa turvaudutaan vanhaan tuttuun asuntojen.hintatiedot.fi-palveluun. Pääkaupunkiseudun kauppoja löydetään saatavilla olevalta viimeiseltä 12 kk:lta seuraavasti: Helsinki (85), Espoo (89), Vantaa (103) ja Kauniainen (7). Muodostetaan vastaavat sijaintikunta-dummyt: D_Helsinki, D_Espoo, D_Vantaa ja D_Kauniainen. Jos vertailukauppa sijaitsee esim. Kauniaisissa, saa muuttuja D_Kauniainen arvon 1 ja muut ko. vertailukauppaan liittyvät dummyt arvon nolla. Lähtötieto-aineistossa sijainti tiedetään postinumeroalueen tarkkuudella, mutta tässä yhteydessä tyydytään (tarkoituksella) kaupunkitason sijaintiin. Tämä tuo tietysti huomattavaa lisähajontaa hintojen vaihteluun eli sen pitäisi laskea mallin selitysastetta selkeästi; näppituntumana 15–20 %. Tarkkaa vaikutusta voisi tietysti lähteä myös selvittämään mutta rajataan tämä kysymys tämän artikkelin ulkopuolelle.

Jatkuviksi muuttujiksi koodataan pinta-ala (area), ikä (age) ja velaton kauppahinta (DfP). Kolmiportaisen kuntoluokituksen (hyvä, tyydyttävä huono) perusteella koodataan dummy-muuttujat D_good (185), D_sat. (82) ja D_poor (17); suluissa kappalemäärät.

Huoneiden lukumäärä koodataan viiteen dummyyn: D_1r (5), D_2r (9), D_3r (34), D_4r (117), D_5r+ (119). Lopuksi on koodattu vielä muuttuja D_sauna (176) ja D_nosauna (108).

Verkossa on tarjolla muutamia ilmaiseksi käytettäviä ohjelmistoja, joista neuroverkkotoiminto löytyy. Vertailematta näitä vaihtoehtoja tässä yhteydessä sen tarkemmin, kohdistamme valinnan WEKA-ohjelmistoon, jota ainakin tutkijat laajalti suosivat. Kyseessä on monipuolisesti toiminnallisuuksia sisältävä analyysiohjelmisto, joka sisältää myös neuroverkkojen laskentamahdollisuuden perusvariaatioilla. WEKA on myös kehitetty varsin helppokäyttöiseksi. Esim. muuttujien skaalaus hoituu automaattisesti, kuten myös edellä selostettu kategoristen muuttujien koodaus dummy-muuttujiksi.

HIUKAN HIENOSÄÄTÖÄ KUITENKIN

Tämäkin ohjelmisto mahdollistaa käyttäjälleen runsaasti mahdollisuuksia tehdä valintoja neuroverkon hienosäädön suhteen. Laskenta voidaan käytännössä suorittaa asetetuilla ja automaattisestikin säätyvillä oletusarvoilla, mutta tehdään kuitenkin muutama ”manuaalinen” säätö.

Yksi tärkeimmistä näistä ns. hyperparametreistä on verkon arkkitehtuurin eli piilokerrosten ja niissä olevien neuronien lukumäärän valinta. Sisääntulokerroksen neuronien lukumäärä tulee annettuna sen mukaan, kuinka monta selittävää muuttu-

jaa verkolle annetaan. Samoin ulostulokerroksessa neuronien lukumäärä määräytyy suoraan sen mukaan, mikä selittäväksi muuttujaksi valitaan. Tässä tapauksessa haluamme selittää hintojen muodostusta, joka saa vain yhdenlaisia jatkuvia arvoja. Näin ollen ulostulokerroksessa on vain yksi neuroni, velaton hinta (DfP).

Piilokerrosten ja siellä olevien neuronien lukumäärän valintaan ei ole olemassa mitään yksiselitteisen optimaalista ratkaisua. Parhaiten toimiva verkko on käyttäjän mielestä toisinaan mahdollisimman nopea, toisinaan mahdollisimman tarkka ja useimmiten näiden ominaisuuksien sopiva yhdistelmä. Valinnan tekemisen voi myös jättää ohjelmistolle, mutta tällä kertaa haluamme määritellä verkon rakenteen itse. Valitsemme verkkoon (peräti) kolme piilokerrosta ja määrittelemme piilokerrosten neuronien lukumääräksi 12,6,3 (neuronien lukumäärän eri kerroksissa ei sinällään tarvitse olla missään tietyssä lukumääräsuhteessa toisiinsa).

Säädettävä ominaisuus on myös "batch size", joka määrittää sen, kuinka usein verkon painokertoimet päivitetään laskennan edetessä. Mikäli "eräkoko" on yksi, päivitetään yhteyksien painot jokaisen vertailukaupan yksittäisen laskemisen jälkeen (eli kun verkko on laskettu eteenpäin ja taaksepäin). Valitettavasti tämän hyperparametrin toimintaa isommilla eräkoolla ei ole ohjelmiston ohjeistuksessa täsmällisesti kuvattu. Pohjimmiltaan kysymys on kuitenkin valinnoista verkon nopeuden ja laskentatarkkuuden välillä.

Yksiselitteisesti parhaita ratkaisuja hyperparametrivalinnoissa on vaikea määritellä etukäteen. Laajoilla aineistoilla laskettaessa puolisenkin tuntia käytetty aika hyperparametrien intuitiiviseen hienosäätöön saattaa maksaa itsensä helposti takaisin varsinaisen laskennan ajansäästönä.

Hyperparametrit *learning rate* ja *momentum* saavat jäädä oletusarvoihinsa (0,3 ja 0,2). Niillä säädellään sitä, kuinka suurina harppauksina yhteyksien painokertoimia muutetaan ja niillä on vaikutuksensa myös laskennan tarkkuuteen.

Sitten päästään lopultakin laskemaan. Aivan ensimmäiseksi testataan verkon laskentanopeutta. Aineiston läpikäyminen 10 000 kertaa (10 000 epookkia) kestää vaivaiset 15 sekuntia. Lähdetään liikkeelle hiukan perusteellisemmin ja säädetään laskenta pysähtymään 100 000 epookin kohdalle. Vajaan kolmen minuutin kohdalla laskenta pysähtyy ja saadaan alla olevassa taulukossa kuvatut tulokset.

Taulukko 1. Laskennan tulostaulu 100 000 epookin jälkeen.

Correlation coefficient	0,9731
Mean absolute error	27 030,5931
Root mean squared error	37 260,6482
Relative absolute error	22,253 %
Root relative squared error	23,2423 %
Total number of instances	284

Poimitaan yllä olevasta taulukosta lähempään tarkasteluun korrelaatiokerroin (correlation coefficient) 0,9731 ja keskivirhe (mean absolute error) 27030,5931. Keskivirhe kertoo, kuinka suuren virheen (positiivisen tai negatiivisen) muodostettu hintamalli keskimäärin laskee yksittäiselle kaupalle. Noin 27 000 euron keskivirhe tuntuu harvat lähtötiedot huomioon ottaen varsin hyvältä tulokselta. Korrelaatiokerroin 97,31 % kertoo samaa tarinaa. Hintamalli pystyy lähes täydelliseen korrelaation todellisten ja laskettujen hintojen kanssa. Korrelaatiokertoimesta laskettu selitysaste on (0,9731)^2 = 94,69 %. Tulokset ovat suorastaan epäilyttävän hyviä.

Lasketaan malli vielä toisilla asetuksilla, jossa laskenta pysäytetään jo 25 000 epookin jälkeen. Tulokset näyttävät edelleen hämmentävän hyviltä. Keskivirhe (MAE) on nyt 37 889,4109 euroa, mutta selitysaste on edelleen niinkin korkea kuin 93,14 %.

LIIAN KOVA OPPIMAAN

Mistä on siis kysymys? Lineaarisella regressioanalyysillä saadaan ani harvoin näin hyviä tuloksia, vaikka käytössä olisivat huomattavasti tarkemmat vertailukauppatiedot. Nythän vertailukaupoista on tiedossa vain muutama hassu ominaisuus, sijaintikin vain kuntatasolla. Relevanttien puuttuvien muuttujien lista on pitkä. Tiedossa ei ole esim. tonttien kokoa eikä mahdollisesti käyttämättä olevaa rakennusoikeutta, jonka arvo pääkaupunkiseudulla voi olla huomattava. Liikenneyhteyksien (matka-aikojen) ja palveluiden välisistä eroistakaan ei ole minkäänlaista tarkempaa tietoa.

Onko neuroverkko siis ylivoimaisen hyvä mallintaja vai miten tulokset pitäisi tulkita? Yhdestä näkökulmasta vastaus on yksinkertainen ja selkeä: neuroverkko on ylivoimaisen hyvä mallintaja. Ongelma on siinä, että se on helposti liiankin hyvä mallintaja. Kyse on pohjimmiltaan neuroverkon perusominaisuudesta, jossa sopivilla hyperparametrien asetuksilla mallista saadaan niin tarkka kuin halutaan. Kun riittävän kauan lasketaan, on mahdollista päästä jopa 100 %:n selitysasteeseen tai ainakin erittäin lähelle sitä. Tämä on mahdollista siksi, että laskennan jatkuessa riittävän kauan verkko alkaa mallintaa paitsi ilmiötä itsessään, myös ilmiöön liittyvää satunnaisuutta.

Kiinteistöjen hinnat määräytyvät vain osittain kaupan kohteen, ympäristön ominaisuuksien yms. tekijöiden perusteella. Hintojen muodostukseen liittyy myös satunnaisuutta, joka riittävän raskaalla laskennalla saadaan tavalla tai toisella mallinnettua. Tästä on kyse myös tämän artikkelin esimerkissä. Hintoihin liittyvä satunnaisuus on tullut mallinnettua samalla itse ilmiötä mallinnettaessa.

Nyt laskettu malli selittää fantastisen hyvin lähtötietoaineiston (training set) hinnat, mutta kun mallilla ryhdytään laskemaan hintoja otoksen ulkopuolelta eli esim. tekemään arviointeja, mallin antamat ennusteet voivatkin olla fantastisen huonoja. Verkko on menettänyt kykynsä yleistää. Ilmiö on nimeltään ylioppiminen (overfitting). Ongelma on merkittävä, sillä ylioppinut verkko menestyy surkeasti varsinaisessa tehtävässään eli laskiessaan arvoja otoksen ulkopuolisille kohteille.

Maankäytön seuraavassa numerossa julkaistavassa kirjoituksen toisessa osassa käydään läpi keinoja ylioppimisen ehkäisemiseksi ja verrataan neuroverkon suorituskykyä lineaarisella regressioanalyysillä laskettaviin tuloksiin. Lisäksi syvennetään keskustelua neuroverkon älykkyydestä.

Lue myös nettiekstra MITEN VERKKO OPPII?