Koneaivoa takomassa

Neuroverkkojen mysteerit auki

Neuroverkot tekevät hämmästyttäviä asioita. Ne tunnistavat ihmisiä kasvonpiirteistä, ohjaavat itsestään ajavia autoja ja tuottavat taidetta. Ne eivät kuitenkaan ole täysin käsittämättömiä taikalaatikoita – kuolevaisellakin on aivan hyvät edellytykset ymmärtää niiden periaatteita.

Teksti: Ville-Matias Heikkilä Kuvat: Tapio Lehtimäki, Mitol Meerna, Ville-Matias Heikkilä

Useimmat tavanomaiset tietojenkäsittelytehtävät eivät kaipaa neuroverkkoja avukseen. Viivanpiirtoa, varmuuskopioiden automatisointia tai toimintapelin mekaniikkaa toteuttava ohjelmoija pystyy analysoimaan ongelman täydellisesti omassa mielessään. Hän voi helposti käydä kaikki mahdolliset poikkeustilanteet systemaattisesti läpi ja järkeillä niille pätevät tai jopa optimaaliset ratkaisut. Kaikki tehtävät eivät kuitenkaan ratkea näin helposti – monien ongelmien kohdalla analyyttinen järki saattaa pettää täysin.

Otetaanpa esimerkiksi pelit. Pelimekaniikka voi olla hyvin helppo toteuttaa, mutta entäpä peliä hyvin pelaava tekoäly? Tekoälyn ohjelmoija päätyy helposti kokeilemaan melko umpimähkäisesti erilaisia parametreja, kaavoja, apumuuttujia ja ehtolauseita. Tällaiseen tilanteeseen jouduttuaan ohjelmoijan onkin usein aiheellista kysyä itseltään, olisiko järkevää kokeilla neuroverkkoa tai jotain muuta automaattisesti säätyvää mekanismia.

Neuroverkon toimintaperiaate

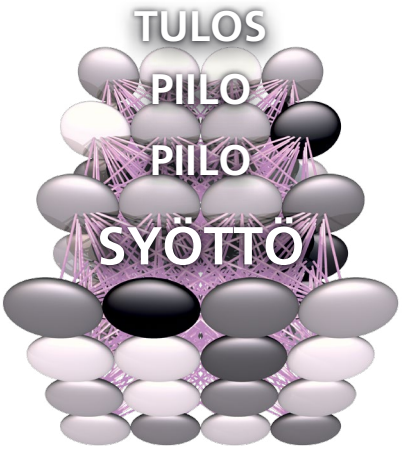
Yksinkertaisia toimintapelejä pelaavat neuroverkot ovat melko helppoja ymmärtää ja toteuttaa, ja ennen kaikkea niiden lopputulokset ovat havainnollisia. Niitä onkin tästä syystä näkynyt monissa viraalivideoissa, jotka tutustuttavat ihmisiä neuroverkkojen periaatteisiin.

Keinotekoinen neuroverkko on hyvin karkeasti määriteltynä matemaattinen malli tai tietokoneohjelma, joka jäljittelee luonnollisia neuroverkkoja, elävien olentojen hermojärjestelmiä. Neuroverkon perusyksiköstä käytetäänkin biologiasta tuttua nimitystä *neuroni*. Yksittäisen neuronin tila kuvataan tietokoneessa yleensä yhdellä liukuluvulla – se on siis huomattavasti yksinkertaisempi kuin elävä hermosolu.

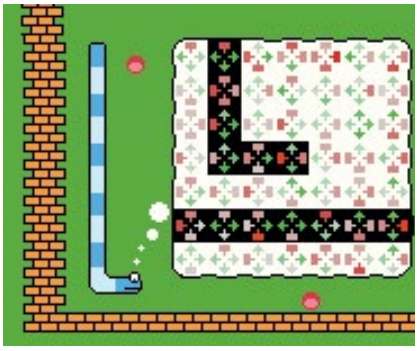
Neuronit on useimmissa verkoissa aseteltu kerroksiksi (*layers*), joista oleellisimpia ovat syöttö- ja tulostuskerrokset. Nokian matopelin kaltaista peliä pelaavalla verkolla voisi olla esimerkiksi 7×7 neuronin syöttökerros, joka esittää madon pään ympäristön siten, että tyhjiä ruutuja vastaa luku 0, esteitä luku 1 ja syötäviä hedelmiä –1. Tulostuskerros puolestaan voisi koostua neljästä neuronista, jotka vastaavat madon neljää mahdollista liikkumissuuntaa. Tulostuskerrosta tulkittaisiin siten, että verkko haluaa madon etenevän siihen suuntaan, jota vastaava neuroni ”loistaa kirkkaimmin”.

Neuronien välillä on linkkejä, joiden biologinen vastine on *synapsi*. Suoraviivaisinta linkitys on ns. eteenpäin syöttävissä verkoissa (*feed-forward network*), joissa yhden kerroksen neuronit linkittyvät aina seuraavan kerroksen neuroneihin ja data kulkee aina yhteen suuntaan.

Kullakin synapsilla on yleensä liukulukuna kuvattu painokerroin, joka ilmaisee sen, kuinka paljon sen lähtöneuronin tila vaikuttaa kohdeneuronin tilaan. Useimmiten neuronin tila-arvoksi asetetaan painotettu summa siihen linkittävien neuronien arvoista. Monesti tulosta vielä ”vahvistetaan” aktivaatiofunktiolla, joka voi olla esimerkiksi maksimi tai tangentti. Aktivaation tehtävä on samantapainen kuin transistoreilla mikropiireissä: se auttaa verkkoa valitsemaan useiden vaihtoehtojen välillä. Painotettua summaa ja



Eteenpäin syöttävän neuroverkon rakenne.



Piilokerrokseton matopeliverkko kytkee pelimaailman ruudut suoraan liikkumissuuntiin.

aktivaatiofunktiota käyttävä neuronirakenne on nimeltään *perseptroni*.

Syöttö- ja tulostuskerroksen välisiä kerroksia kutsutaan piilokerroksiksi (*hidden layers*). Niitä ei aivan kaikissa tehtävissä tarvitse, mutta tällöin on kyse pikemminkin lineaarifunktiosta kuin varsinaisesta neuroverkosta.

Matopeliä jotenkuten pelaavan verkon saa toimimaan ilman piilokerroksiakin. Tällöin verkon voi kuvata yksinkertaisesti synapsitaulukkona, jossa on kullekin pelinäkymän pikselille neljä kerrointa. Nämä kertoimet osoittavat, kuinka paljon madon etenemistodennäköisyys kuhunkin suuntaan muuttuu siitä, jos kyseisessä ruudussa on jotain muuta kuin tyhjää. Aiemmin mainituilla syöttö- ja tulostuskerroksen neuronimäärillä synapsimääräksi tulee kaikkiaan 7×7×4 eli 196, mikäli kerrokset kytketään toisiinsa kokonaan.

Se oppii itse!

Vaikka neuroverkkojen painokertoimia voi säätää käsinkin, on niiden varsinainen juju säädön automatisoinnissa eli verkon koulutuksessa.

Koulutustavat voidaan jakaa ohjattuihin (*supervised*) ja ohjaamattomiin (*unsupervised*). Ohjattua tapaa voidaan käyttää, jos tarjolla on runsaasti ”esipureksittua” dataa, jossa esimerkkisyötteet on yhdistetty haluttuihin tulosteisiin. Peliä pelaavassa verkossa tällaisena datana voitaisiin käyttää esimerkiksi hyvien ihmispelaajien reaktioita erilaisiin pelitilanteisiin. Ohjaamattomassa verkossa valmiita ei esimerkkejä ole.

Pelejä pelaaviin verkkoihin sopiva ohjaamaton koulutustapa on *neuroevoluutio*. Siinä annetaan satunnaisten verkkojen kilpailla keskenään, ja kunkin kierroksen jälkeen huonoiten pärjänneet verkot korvataan uusilla verkoilla, jotka on risteytetty parhaiten pärjänneistä. Useiden tuhansien risteytyssukupolvien kuluessa verkkojen pitäisi vähitellen säätyä aina vain paremmin tehtävän hallitseviksi. Neuroevoluutio on helppo toteuttaa ja siksi sopiva vaikkapa harjoitusprojektiksi. Esimerkkikuvan matopeliverkko on treenattu evoluutiolla.

Eteenpäin syöttävät verkot koulutetaan useimmiten takaisinvälityksellä (*back-propagation*). Tässä tekniikassa verkolle annetaan esimerkkisyöte ja verrataan sen antamaa tulostetta haluttuun tulosteeseen. Tämän jälkeen lasketaan kerros kerrallaan tulostuskerrokselta taaksepäin edeten, kuinka paljon kunkin synapsin painokerrointa on rukattava ja mihin suuntaan, jotta tulos olisi hieman lähempänä haluttua. Matemaattisemmin ilmaistuna tämä on gradientin seuraamista, ja siihen yleisesti käytetty algoritmi on nimeltään SGD eli *stochastic gradient descent*.

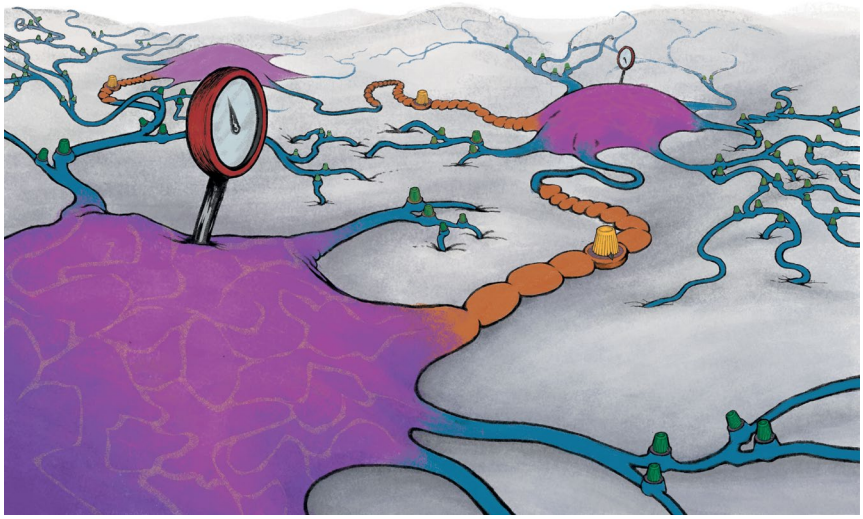
Gradienttia orjallisesti seuraamalla juututaan helposti ns. paikallisiin maksimeihin: ei päästä mäennyppylältä sen vieressä olevalle vuorelle, koska välissä on alamäkeä. Ongelmaa voi ratkaista satunnaistekijöillä. SGD:ssä satunnaisuutta tuo koulutusdatan läpikäyntijärjestyksen arpominen, ja lisäksi verkon neuroneista voidaan kytkeä tietty satunnainen osuus pois päältä kullakin kierroksella.

Verkko voi luoda kategoriansa myös itse, jolloin se soveltuu esimerkiksi tavallisen ja epätavallisen erottamiseen toisistaan. Teuvo Kohosen itseorganisoituvat kartat toimivat näin. Kaikissa neuroverkoissa ei myöskään ole erillistä koulutus- ja käyttömoodia, vaan ne oppivat käytön aikana samaan tapaan kuin biologiset hermostot. Tämäntyyppiset verkot koostuvat muistavista neuroniyksiköistä, joiden tehtävänä on oppia ennustamaan niihin tulevia signaaleja. Jos ennuste ei vastaa todellisuutta, yksikkö voi vaikkapa ”ylättyä” ja antaa siitä signaalin eteenpäin.

Kuvantunnistusverkot

Hahmontunnistus on ongelmatyyppi, jossa neuroverkot pääsevät loistamaan. Kuvantunnistusverkoilla on myös äskettäin havaittu häkellyttäviä graafisia kykyjä, kun niiden on annettu tuottaa takaisinvälityksen kautta omia ”hallusinaatioitaan”.

Eräänlainen kuvantunnistusohjelmistojen vuotuinen kuninkuusmittelö on ILSVRC (*Imagenet Large Scale Visual Recognition Challenge*). Kukin kilpailuun osallistuva ryhmä toimittaa siihen ohjelman – tyypillisesti esikou-



Millä tavoin synapsien säätönuppeja on pyöriteltävä, jotta mittarit osoittaisivat aina oikein? Ihmiseltä saattaisi mennä hermot, joten tehtävä on paras jättää koneelle.

VGG-tiimin 16-kerroksinen verkko vuodelta 2014.

lutetun neuroverkon – jonka tehtävänä on määrittää sille annetuista kuvista, mitä niissä on. Luokittelukategorioita on kaikkiaan tuhat erilaista, ja niihin kuuluu mm. satoja eläinlajeja ja arkielämän esineitä.

Viime vuosina hyvin menestyneet verkot ovat usein olleet ns. syviä konvolutiivisia verkkoja. Syvyys tarkoittaa sitä, että verkossa on paljon piilokerroksia – jopa useita kymmeniä. Tämä antaa verkolle tilaa muodostaa varsin pitkälle vietyjä abstraktioita näkemästään. Konvolutiivisuus puolestaan tarkoittaa kytkentätapaa, jossa kerroksen kuhunkin neuroniin vaikuttaa edellisessä kerroksessa samassa kohti oleva 3×3, 5×5 tai useamman neuronin tai neuronipinon rypäs. Tällä tavoin kytketyt kerrokset toimivat pitkälti samaan tapaan kuin kuvankäsittelyohjelmien konvolutiiviset suotimet.

Monet ILSVRC:ssä kilpailleet verkot ovat vapaasti ladattavissa ja käytettävissä ei-kaupallisiin tarkoituksiin. Googlen ja Oxfordin yliopiston menestyksekkäät verkot ovat ajettavissa *Caffe*-ohjelmistolla, joka on etenkin syvien verkkojen tutkimukseen kehitetty avoimen lähdekoodin ohjelma. Caffe panostaa nopeuteen ja osaa hyödyntää CUDA-rajapinnan kautta mm. grafiikkasuorittimien rinnakkaislaskentaa. Ohjelmointirajapintoja on tarjolla C++:lle ja Pythonille, ja etenkin Python-esimerkkikoodia on varsin helppo löytää.

Ohessa on kuvattu Oxfordin yliopiston Visual Geometry Groupin, vuonna 2014 käyttämä 16-kerroksinen verkko, joka yksinkertaisesta rakenteestaan huolimatta pärjäsi kisassa erinomaisesti. Useimmat sen kerrokset ovat 3×3-konvoluutiokerroksia, jotka leikkaavat samalla summista negatiiviset arvot nolliksi. Aina muutaman konvoluutiokerroksen jälkeen on *max-pool*-kerros, joka ottaa kustakin 2×2:n tai 3×3:n neuronin ryppäästä talteen suurimman arvon. Maxpool-kerrokset vähentävät verkon herkkyyttä kuvien merkityksettömille asemointieroille. Viimeistä maxpool-kerrosta seuraa kolme kerrosta, joiden jokainen neuroni on kytketty jokaiseen edellisen kerroksen neuroniin. Näistä viimeisessä on tuhat neuronia – yksi kutakin luokitteluryhmää varten.

Verkon vaatima tallennustila riippuu varsin suoraan sen synapsimäärästä. Googlen kuvantunnistusverkko Googlenetin mallidata vie levyltä reilut 50 megatavua, kun taas VGG:n verkkojen koot ovat olleet useita satoja megoja. Muistiin ladattu verkko vaatii lisäksi tilaa neuronikohtaiselle datalle.

Mysteereihin valoa uneksimalla

Neuroverkkojen rakenteita ylläpitävä algoritmiikka on varsin yksinkertaista, ja sen voi halutessaan ohjelmoida melko kivuttomasti itsekin. Se kuvaa kuitenkin pelkän raaka-aineen, josta varsinainen logiikka muotoutuu koulutuksessa. Tämä logiikka on huomattavasti vaikeaselkoisempaa, optimointialgoritmit kun eivät kunnioita ihmisinsinöörien modularistisia ihanteita. Sitä on ollut vaikea kuvata asiantuntijoidenkaan ymmärtämässä muodossa.

Jos kuvantunnistusverkon tietyn kerroksen neuronitilat visualisoidaan suoraan 2D-kuvaksi, tuloksena on yleensä epämääräisiä läikkiä. Nämä läikät ovat kuin aivojen magneettikuvia – tutkija saattaa pystyä esittämään niiden perusteella valistuneen arvauksen, mitä verkossa tapahtuu, mutta maallikolle ne eivät sano mitään. Huomattavasti havainnollisempi visualisointitapa on Googlen tutkijoiden melko äskettäin keksimä, takaisinvälitysmekanismia hyödyntävä menetelmä.

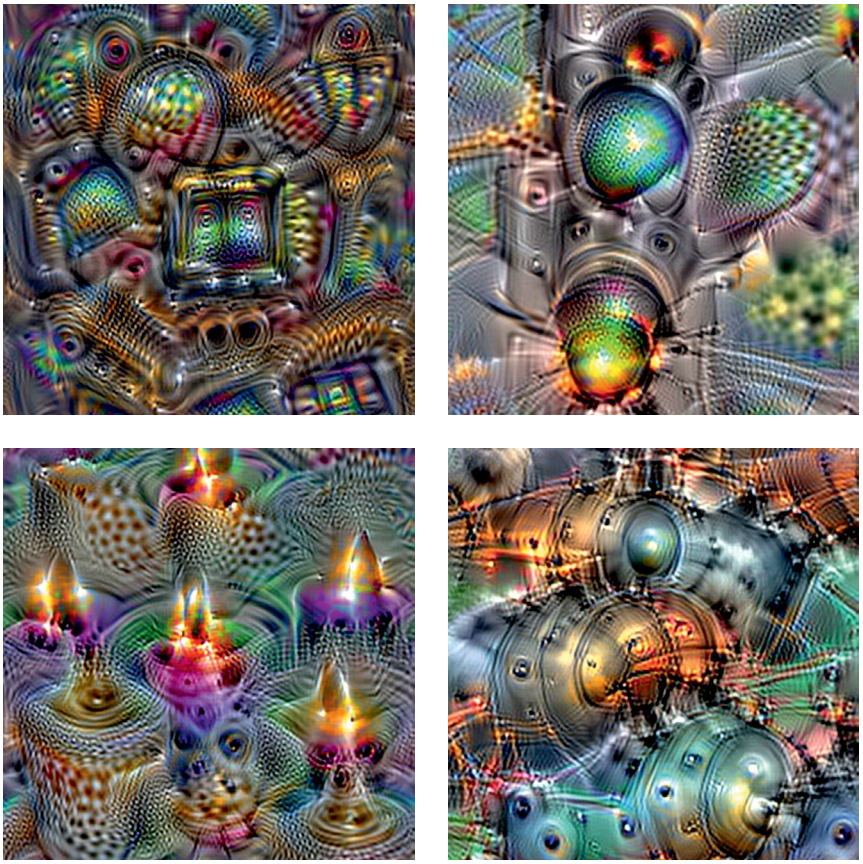
Miltä tietokone näyttää neuroverkon mielestä? Syötetään verkolle aluksi kohinaa, josta se saattaa tunnistaa mitä tahansa. Tämän jälkeen lasketaan, kuinka kuvaa pitää muuttaa, jotta sen tunnistus olisi hieman lähempänä tietokonetta. Pöytätietokoneen tunnistava tuloskerroksen neuroni on ILSVRC12-kategorioiden mukaan numero 527, joten sen ulostulo on saatava kasvamaan.

Koulutuskäytössä takaisinvälitykseen kuuluu kaksi vaihetta: ensin lasketaan, kuinka kunkin edelliskerroksisen neuronin ulostulo muuttuu, jos tuloskerroksen neuroneille tehdään halutut muutokset, ja sen jälkeen rukataan synapsikertoimia näiden muutosten mukaan. Tässä tapauksessa jälkimmäinen vaihe voidaan unohtaa, sillä meitä kiinnostaa vain, kuinka syöttökerroksen neuronien tilat muuttuvat tuloskerroksen muutoksen myötä. Kun muutoksia kasataan silmukassa alkuperäisen kuvan päälle, kohinan seasta paljastuu vähitellen verkon käsitys tietokoneesta.

Entäpä, jos halutaan tietää, mitä kuvantunnistusverkon mikäkin kerros tekee? Kun tietyn kerroksen neuronien arvoja kasvatetaan sitä enemmän, mitä suurempia ne ovat ennestään, saadaan takaisinvälityksen avulla vahvistettua kuvasta niitä yksityiskohtia, joihin tämä nimenomainen kerros kiinnittää huomiota. Alimmilta kerroksilta paljastuu tyypillisesti perustason kuvankäsittelyoperaatioita kuten värirajojen etsintää, kun taas ylimmät kerrokset tuottavat jo selvästi tunnistettavia eläimiä, esineitä ja rakennuksia. Tämä kuvan piirteiden vahvistamistapa sai Googlen tutkijoilta nimen *deep dream*.

Taiteelliset ulottuvuudet

Googlen kuvantunnistusverkon tuottamat deep dream -kuvat saivat kesällä 2015 aikaan netti-ilmiön, jossa samaa



VGG:n verkolta maaniteltiin esiin kuvia, jotka sen mielestä tunnistuvat erityisen vahvasti tiettyihin aiheisiin: hedelmäpeli, liikennevalo, kynttilä, höyryveturi.

Tällaisia pintoja syntyy, kun VGG:n verkolle annetaan oppaaksi kuvat lahottajasienistä ja avaruusaluksesta.

Googlen esimerkkikoodia sovellettiin sellaisenaan kaikenlaisiin kuviin. Koska ILSVRC:n tuhanteen tunnistuskategoriaan kuuluu useita eri koirarotuja, syntyy Imagenet-datalla treenatuille verkoille herkästi taipumus etsiä kuvista yksityiskohtia, jotka erottavat näitä rotuja toisistaan. Niinpä deep dream -kuviinkin generoitui koiranpäitä suorastaan kyllästymiseen asti.

Kuvantunnistusverkoissa on kuitenkin huomattavasti enemmän taiteellista potentiaalia, kuin mihin kesän järkytysmeemi ehti yltää. Yksityiskohtien vahvistumista voi ohjata lukemattomilla eri tavoilla, verkkoja voi treenata omilla materiaaleilla ja niitä voi yhdistää moniin muihin tekniikoihin. Mahdollisuuksissa on hädin tuskin päästy edes raapaisemaan pintaa.

Hyvä tapa saada aavistus verkon luovuudesta on antaa sen ”uneksia” eli sukeltaa omien tuotostensa sisään: kun kuvan yksityiskohtia on vahvistettu deep dream -algoritmilla, sitä skaalataan hieman ja skaalattu kuva syötetään takaisin algoritmille. Kun tätä jatketaan silmukassa, syntyy fraktaalizoomeria muistuttava animaatio, jossa kuvan yksityiskohdista paljastuu loputtomiin uusia yksityiskohtia.

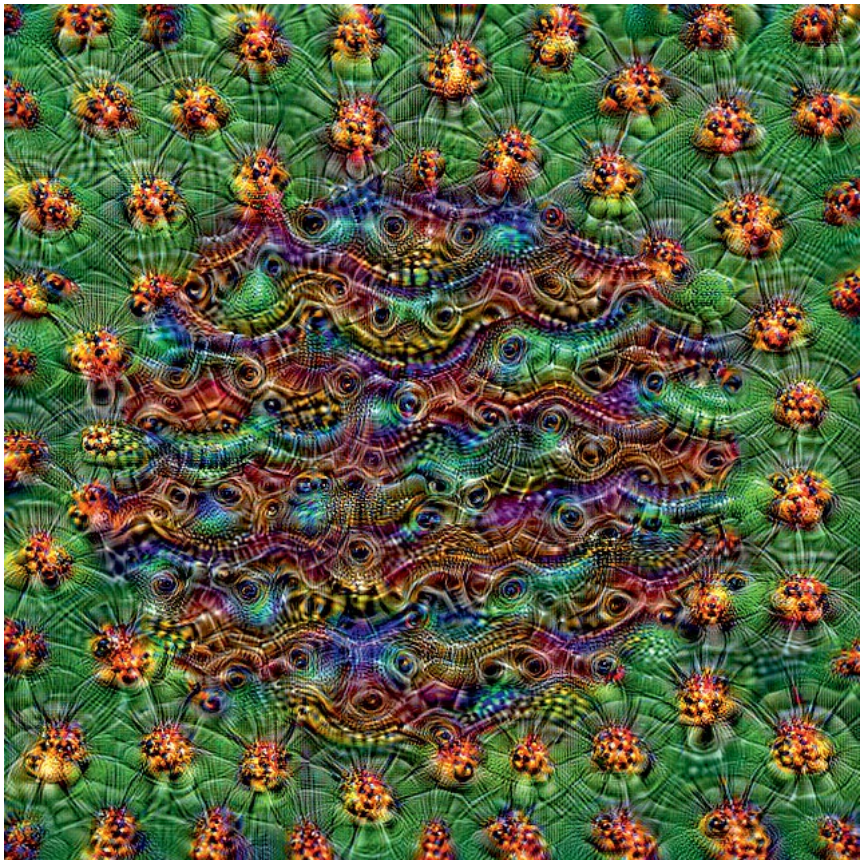
Vapaa uneksinta paljastaa kuitenkin vain pienen osan verkon mielikuvitusmaailmasta, sillä tietyt elementit korostuvat siinä suhteettoman paljon. Kuten ihmismieli on kärkäs tunnistamaan kasvoja jopa epämääräisistä läikistä, myös neuroverkot kehittävät herkästi vastaavanlaisen taipumuksen. Unien sisältöön voi vaikuttaa rukkaamalla kaavoja, jotka määräävät, minkä neuronien arvoja voimistetaan ja kuinka painokkaasti.

Helppo tapa ohjata uneksintaa on antaa verkolle opaskuva (*guide*), jossa on halutunlaisia piirteitä. Opaskuvan tuottamia neuroniarvoja käytetään uneksinnan aikana painokertoimina neuroniarvojen muutoksille. Näennäisestä hallittavuudestaan huolimatta tulokset saattavat kuitenkin olla hyvin odottamattomia: esimerkiksi eräs avaruusalusta esittävä opaskuva tuottaa gigermäistä lonkeroseinämää. Avaruusalus ei kuulu Imagenet-kategorioihin, joten verkko ilmeisesti yhdisti kuvan merenelämiin.

Yleiset kuvantunnistusverkot ovat uneksintakokeilujen perusteella erityisen kyvykkäitä pienissä orgaanisissa yksityiskohdissa, jotka ovat toisinaan jopa häiritsevän realistisia. Pinnat, varjostukset ja palasten yhteenliittämiset ovat verkkojen vahvuusalueita, kun taas perspektiivit ja rakennetut muodot ovat ainakin Imagenet-kuvista oppinsa saaneille verkoille paljon vaikeampia. Kokeilemistani esitreenatuista verkoista ainoastaan Googlenet tuottaa uneksiessaan maisemakuvia, ja niissäkin saattaa kuvan alareunaan ilmestyä toinen taivas.

Deep dream -tyyppiset algoritmit saattavat hyvinkin ruveta valtaamaan alaa taiteilijoiden apureina, jotka väkertävät yksityiskohtia ja varjostuksia. Taiteilija saattaa piirtää esimerkiksi luonnoksen, jossa on eri alueet merkitty sen mukaan, mitä niille halutaan tehtävän. Reaaliaikaisemmassa vaihtoehdossa nämä eri käsittelytavat voisivat olla erilaisia ”siveltimiä”. Lisäksi esimerkiksi äskettäisessä Siggraph-konferensissa oli artikkeli kuvan valaistuksen muuttamisesta neuroverkoilla. Mahdollisuuksia on siis moneen suuntaan.

Nykyisten kuvantunnistusverkkojen yhdellä kertaa käsittämä näkökenttä on yleensä vajaat 256×256 pikseliä.



Kuvan keskiosaan on uneksittu sisältöä eri opaskuvalla kuin reunoille. Rajakohdassa näkyy, kuinka verkko pyrkii sovittamaan erityyppisiä rakenteita yhteen.

Network in Network -verkon rakenne.

Kuvan voi toki rakentaa useissa palasissa ja skaalaustasoissa, mutta tällöin eri vaiheet saattavat ikävästi sotkea toisiaan. Myös osa tämän artikkelin kuvista on tästä syystä varsin karkeita.

Omaa verkkoa opettamaan

Esitreenatuilla verkoilla on mukava leikkiä, mutta silloin ollaan sidoksissa niihin kuviin ja kategorioihin, joilla verkko on alun perin koulutettu. Tästä syystä olisi mukava kouluttaa aivan oma verkko tai vaikka useampikin, joiden taiteelliset taipumukset ovat yhtä persoonalliset kuin kouluttajansakin.

Koska monimutkaisten verkkojen treenaukseen saa helposti uppoamaan paljon laskentaresursseja, lienee aiheellista vältellä sooloilua ja tukeutua valmiisiin malleihin. Tutkijat kuitenkin ovat vuosikausien ajan viilailleet kuvantunnistusverkkojaan.

Mutta mitkä mallit ovat hyviä taidekäyttöön? Hyvä menestys ILSVRC:n tunnistustehtävissä ei välttämättä tarkoita, että sen tuottama takaisinvälityskuvasto olisi mitenkään esteettistä. Latailin vertailun vuoksi Caffe-wikin *model zoosta* joukon erilaisia verkkoja, jotka oli valmiiksi koulutettu samalla Imagenet-datalla, ja annoin jokaisen uneksia jonkin aikaa samalla opaskuvien sarjalla. Tein ajoista myös Youtube-videon (*Comparing the hallucinations of four similarly trained neural nets*).

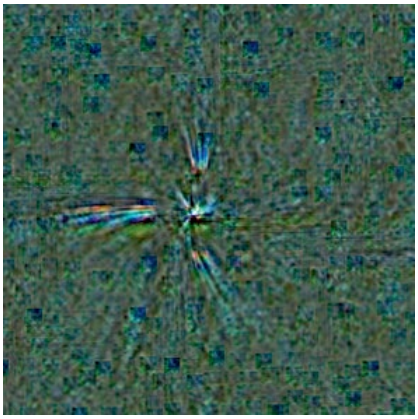
Vaikka Googlenet ja VGG:n CNN-S-verkko tuottivatkin esteettisesti mielenkiintoisinta jälkeä, päätin valita ensimmäiseksi koulutettavakseni Singaporen yliopistossa kehitetyn NIN-mallin (*Network in Network*). Pääasiallisena syynä valintaani oli resurssinkäyttö: verkkoa on nopeampi ajaa kuin kilpailijoitaan, ja sen myös luvataan oppivan nopeasti. Käytössäni ei ole CUDAn tukemaa laskentarautaa, enkä halua odotella ensikokeilujeni tuloksia kuukausitolkulla.

NIN poikkeaa tämänhetkisestä kuvantunnistusverkkojen valtavirrasta siinä, että se käyttää lineaaristen konvoluutiosuotimien sijaan pieniin neuroverkkoihin pohjautuvia suotimia. Singaporelaiset uskovat, että pienoisverkkojen ilmaisuvoima lineaarisiin funktioihin nähden antaa NIN-verkolle kokoonsa nähden vahvan abstraktiokyvyn, ja se päihittääkin tunnistustehtävissä monet isommat verkot.

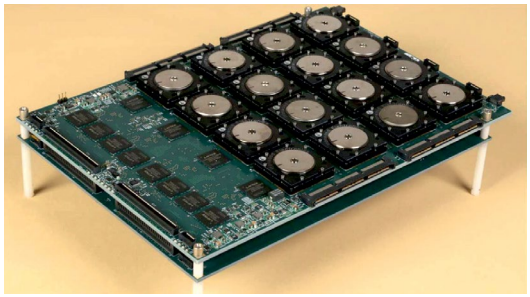
Kuvantunnistusverkon ohjattu koulutus tarvitsee valtavan määrän luokiteltua kuvamateriaalia. Imagenet tarjoaa yksinkertaisen tekstitiedoston, jossa on hulppeat 14 miljoonaa kuvaurlia yhdistettynä kymmeniintuhansiin Wordnet-indekseihin, jotka toimivat luokittelujen pohjina. Imagenetin luokitteluissa on silti ammottavia aukkoja – en esimerkiksi löytänyt siitä havupuita lainkaan. Tein siksi sen avuksi skriptin, joka täydentää aukkoja Flickr-hakujen avulla.

Ajattelin aloittaa kevyesti opettamalla verkon tunnistamaan muutamaa eri sientä ja marjaa – kaikkiaan noin 12 000 kuvan joukko. Valikoimaa voisi laajentaa myöhemmin muunlaisiin kohteisiin, jos tulokset näyttävät lupaavilta.

Kun treenauskuvat on haettu tiedostoihin, niistä muodostetaan *LMDB*-tietokanta Caffen mukana tulevalla *convert_imageset*-työkalulla. Avuksi tarvitaan myös tekstitiedosto, jossa on kullakin rivillä tiedoston nimi ja sen



NIN-verkon unet eivät ole vielä sadantuhannen koulutusiteraationkaan jälkeen kovin tolkullisia.



DARPAn SyNAPSE 16 -neurolaskentalauta mallintaa rinnakkain 16 miljoonaa neuronia ja 4 miljardia synapsia. Kuva: DARPA

sisällön luokitteleva kategorianumero välilyönnein erotettuna. Samalla työkalulla kannattaa muodostaa myös testauksessa käytetty suppeampi *val*-kuvasto, jolla verkon tunnistuskyvyn kypsymistä voi seurata. vMolemmista kuvastoista lasketaan *compute_image_mean*-työkalulla keskiarvo, joka vähennetään kaikista verkolle syötettävistä kuvista. Jos keskiarvokuvaa ei lasketa, verkon parametrit voivat ruveta koulutusvaiheessa puoltamaan äärettömyyksiä päin, jolloin koko ajo epäonnistuu.

Caffe-oppaat neuvovat yleensä kouluttamaan verkot puhtaalta pöydältä. Tässä on kuitenkin omat ongelmansa: suuri osa laskenta-ajasta menee siihen, kun verkko opettelee näkemisen alkeet, ja jos koulutus menee pieleen epäonnisten valintojen vuoksi, verkko ei koskaan opi edes niitä. Siksi voi olla mielekästä käyttää pohjana esitreenattua verkkoa. Jos verkossa ennestään oleva kuvasto häiritsee, kouluttaja voi alustaa verkon kerroksia yksi kerrallaan ylhäältä alaspäin, kunnes se ei enää uneksi mitään tunnistettavaa. Valmiin verkon täydennys- ja uudelleenkoulutuksessa on kuitenkin omat hankaluutensa siinäkin.

Neuroverkon luominen on hyvin erilaista kuin perinteinen ohjelmistonkehitys. Sitä voisi verrata vaikkapa siihen, mitä kasvien kasvattaminen on verrattuna koneiden rakentamiseen. Kouluttaja tarjoaa verkolle sopivaa kasvualustaa ja ravintoa ja ehkä välillä hieman karsii tai kitkee. Toki neuroverkkojakin voi rakentaa kouluttamalla sen osia erikseen ja liittämällä niitä yhteen, mutta etenkin ns. syväoppimiseen (deep learning) kuuluu se, että myös ison mittakaavan rakenteet muotoutuvat verkkoon ilman insinöörityötä.

Lähemmäksi ihmisaivoja

Tavanomaiset eteenpäin syöttävät verkot ovat sielukkaista taipumuksistaan huolimatta aika paljon lähempänä tietokoneohjelmia kuin ihmisaivoja. Kun verkko on kertaalleen koulutettu, se antaa samalla syötteellä joka kerta saman tulosteen – ellei siihen ole varta vasten rakennettu satunnaisuutta. Verkon kehitystyö ja käyttö ovat selvästi erillisiä asioita, eikä koulutukseen tarvitse välttämättä käyttövaiheessa palata enää koskaan. Syväkin eteenpäin syöttävä verkko on oikeastaan ”tyhmempi” kuin tietokone: sen voi rakentaa vaikkapa analogiseksi piiriksi, johon ei kuulu yhtäkään Turing-täydellistä osaa.

Biologiset hermostot eroavat yksinkertaisimmista neuroverkkomalleista etenkin siinä, että ne ovat luonteeltaan ajallisia. Ne reagoivat muuttuviin tilanteisiin ja sopeutuvat ajallisiin signaalinmuutoksiin. Keinotekoiseen neuroverkkoon saa aikaulottuvuuden esimerkiksi linkittämällä joitakin sen neuroneita edellisen ajokierroksen neuronitiloihin, jolloin syntyy *rekurrentti* verkko (RNN). Toinen vaihtoehto on antaa verkolle muistisoluja, joiden sisältöä se pystyy lukemaan ja kirjoittamaan.

Ajalliset verkot soveltuvat toisaalta esimerkiksi puheen, musiikin tai tekstin tunnistamiseen ja tuottamiseen, toisaalta taas ohjaustehtäviin, joita on esimerkiksi robotiikassa ja teollisuudessa. Ajalliset verkot vaativat kuitenkin usein toisenlaisia koulutusmenetelmiä, koska suhde syötteen ja tulosteen välillä ei ole enää niin suoraviivainen.

Kun verkolla on oma muisti, joka pystyy korvaamaan synapsipainotusten tehtäviä, se ei välttämättä tarvitse erillistä koulutusvaihetta vaan oppii toimiessaan. Tällaisia verkkotyyppejä ovat esimerkiksi LSTM (*Long Short-Term Memory*) ja PBWM (*Prefrontal Cortex Basal Ganglia Working Memory*), joista jälkimmäisen esikuvana on varsin suoraan ihmisaivojen otsalohko. Nämä rakenteet pystyvät jäljittelemään yhtä lailla perinteisen tietokoneen kuin biologisten aivojenkin toimintaa.

Erilaisia neuroverkkotyyppejä on lukemattomia, ja rajatapauksissa voi olla epäselvää, milloin kyseessä on neuroverkko ja milloin ei. Esimerkiksi Googlen kääntäjä on tilastollinen itseoppiva järjestelmä samaan tapaan kuin monet neuroverkot, mutta sen tallennusmalli ei perustu synapsityyppisiin säätyviin parametreihin, joten sitä ei pidetä neuroverkkona.

Erityyppisten verkkojen toteuttamiseen on tarjolla erilaisia ohjelmistopaketteja. Esimerkiksi Caffe on tarkoitettu erityisesti syväoppiviin eteenpäin syöttäviin verkkoihin, vaikka sen pystyykin jotenkuten taivuttamaan myös rekurrentteihin ja muistaviin verkkoihin. Caffe kuitenkin laajentaa jatkuvasti alaansa, joten sen opettelu ei ole välttämättä mikään huono päätös aivosimulaatiostakaan kiinnostuneille.

Laitteistopuolella neurolaskentaan käytetään useimmiten joko tavallista suoritinta tai GPU:ta, mutta erityisesti sitä varten suunnitellun laskentalaitteiston voi olettaa yleistyvän tulevaisuudessa. Nanometriluokan ”neuromorfisten” piirien kehittämiseksi on käynnissä useita tutkimusprojekteja mm. HP:llä ja IBM:llä.

Neurolaskenta ei kuitenkaan vaadi erikoista rautaa. Esimerkiksi moniin mobiilisovelluksiinkin kuulu neuroverkkopohjaisia hahmontunnistimia. Samalla Googlen kaltaiset tahot kyttäävät ihmisiä yhä laajemmalla koneistolla, jossa neuroverkot ovat yhä tärkeämmässä osassa. Tämän vuoksi tavallisenkin kansalaisen on hyvä oppia ymmärtämään neuroverkkoja, vaikkapa sitten kuvantunnistusverkoilla leikkimällä. Ja kun leikkiessä saa samalla näkökulmaa myös omien aivojensa toimintaan, niin kovin hyviä syitä jättää tutustumatta neuroverkkoihin ei enää ole.