

Andreas Luoto

Neuroverkkoihin perustuva ja kovakoodattu tekoäly

Tietotekniikan Kandidaattitutkielma

2. elokuuta 2023

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Andreas Luoto

Yhteystiedot: andreas.al.luoto@student.jyu.fi

Ohjaaja: Tytti Saksa

Työn nimi: Neuroverkkoihin perustuva ja kovakoodattu tekoäly

Title in English: Neural network-based and hard-coded artificial intelligence

Työ: Kandidaattitutkielma

Opintosuunta: Kaikki opintosuunnat

Sivumäärä: 22+0

Tiivistelmä: Tämä kirjoitelma on tehty kirjallisuuskatsauksena. Sen tavoitteena on tutkia kovakoodatun tekoälyn sekä neuroverkkoja hyödyntävän tekoälyn eroja ja miksi esimerkiksi shakkia käytetään usein niitä kehitettäessä. Tutkielmassa käsiteltyjen artikkelien pohjalta neuroverkot ovat todennäköisimmin tulevaisuuden tekoälyn pohja, mutta kovakoodattua tekoälyä ei tule sulkea pois kaikissa tilanteissa.

Avainsanat: Tekoäly, kovakoodaus, neuroverkot

Abstract: This essay has been written as a literature review. Its aim was to examine the differences between hard-coded artificial intelligence and artificial intelligence that utilizes neural networks, and why, for example, chess is often used when developing them. Based on the articles I have read, neural networks are likely to be the foundation of future AI, but hard-coded AI should not be excluded in all situations.

Keywords: Artificial intelligence, hard-coding, neural networks

Kuviot

Kuvio 1. Yksinkertainen kolmitasoinen neuroverkko (O'Shea ja Nash 2015) neuro- verkko kuva	10
---	----

Sisällys

1	JOHDANTO	1
2	TEKOÄLYN HISTORIA.....	3
	2.1 Shakki osana tekoälyn kehitystä.....	3
	2.2 Deep Bluen toiminta ja merkitys	4
	2.3 AlphaZeron toiminta ja merkitys	5
3	KOVAKOODATTU TEKOÄLY	7
	3.1 Kovakoodattu tekoäly yleisesti	7
	3.2 Kovakoodatun tekoälyn toiminta ja käyttötarkoitus	8
4	NEUROVERKKOIHIN POHJAUTUVA TEKOÄLY	9
	4.1 Neuroverkot yleisesti.....	9
	4.2 Syötteet, painot ja aktivaatiofunktiot	10
	4.3 Toiminta ja käyttötarkoitus	11
5	TEKOÄLYJEN VERTAILU	13
	5.1 Tekoälyjen vaikutus tulevaisuudessa	13
	5.2 ChatGPT.....	14
6	YHTEENVETO.....	15
	LÄHTEET	16

1 Johdanto

Tekoälyn alkuperäisenä tarkoituksena oli luoda kehittyneitä sekä monimutkaisia järjestelmiä, jotka päihittäisivät ihmisen kaikissa tilanteissa (Poola 2017). Tekoäly on ja tulee luultavasti olemaan tulevaisuudessa iso osa ihmisten jokapäiväistä elämää, riippuen suuresti siitä miten ihmiset kykenevät ratkaisemaan eri eettiset, oikeudelliset sekä sosiaaliset kysymykset (Floridi 2019). Tämä on hyvin tärkeää ottaa huomioon varsinkin silloin, kun esimerkiksi automaattiohjattavia autoja tulee lisää liikenteeseen. Kuka on vastuussa, jos tekoälyä ajaessaan hyödyntävä itseohjautuva auto vahingoittaa ihmistä?

Tässä tutkielmassa vertaillaan neuroverkkoja käyttäviä tekoälyjä ja kovakoodattua tekoälyä hyödyntäviä ohjelmia. Tutkielmassa käsitellään myös muun muassa sitä, miksi juuri shakkia hyödynnetään tekoälyä kouluttaessa ja ohjelmoidessa. Kaksi yleisesti tunnettua esimerkkiä shakkia pelaavista tekoälypohjaisista ohjelmista ovat Deep Blue ja AlphaZero. Tutkielmassa käsitellään juuri kyseistä AlphaZeroa, joka hyödyntää useita päällekkäisiä neuroverkkoja tekoälyssään, sekä vanhaa kovakoodattua Deep Bluesta, joka pelasi kuuluisimman shakkipelinsä vuonna 1997 sen ajan suurinta shakkimestaria vastaan (Hassabis 2017). Deep Bluen pelityylin ollessa enemmän tietokonemainen pelaaja, AlphaZeron pelityyli muistuttaa enemmän ihmismäistä pelaamista. Sen johdosta onkin leikillään sanottu romanttisen shakin aikakauden palanneen (Klein 2022).

Koska oppiva tekoäly ei tarvitse valmista dataa, sen voi opettaa tekemään paljon erilaisia tehtäviä asiakaspalvelusta sairauksien diagnosoimisiin. Kovakoodatussa tekoälyssä ohjelma suorittaa usein vain tiettyä asiaa ennalta määrätyillä ohjeilla, jolloin se ei sovelle ollenkaan suorittaessaan tehtäviään. Oppivaa tekoälyä voitaisiin pitää lähtökohtaisesti parempana kuin kovakoodattua tekoälyä perustuen sen kykyyn oppia ympäristöstään ja omista ratkaisuisistaan, kuten kappaleessa aiemmin pohdittiin. Tutkielmassa selvitetään, onko kovakoodattu tekoäly tosiaan huonompi ratkaisu kaikissa tilanteissa, vai onko molemmilla toteutustavoilla paikka ihmisten tulevaisuuden tekoälyn hyödyntämisessä.

Tässä tutkielmassa vertaillaan kovakoodattua tekoälyä neuroverkoilla toimivaan tekoälyyn. Luvussa kaksi mietitään, miksi shakkia käytetään tekoälyjen tehokkuuksia testatessa. Tä-

män lisäksi käydään läpi tekoälyn historiaa kahden eri tekoälyohjelman näkökulmasta. Kolmannessa luvussa avataan enemmän kovakoodattua tekoälyä yleisellä tasolla ja selitetään tarkemmin sen toiminnasta. Luvussa neljä perehdytään neuroverkkoihin sekä niiden toimintaan ja käyttötarkoitukseen. Viidennessä luvussa pohditaan tekoälyn roolia ihmiskunnan tulevaisuudessa, sekä kerrotaan hieman uudesta ChatGPT tekoälystä. Lopuksi tutkielmassa on yhteenveto aiheista, joita käsitellään tässä kirjallisuuskatsauksessa.

2 Tekoälyn historia

Tekoäly on ollut esillä jo vuosikymmenien ajan esimerkiksi lääketieteessä, mutta vasta lähihistoriassa sitä on pystytty kehittämään vielä tehokkaammaksi eri menetelmien avulla. Esimerkiksi jo 70-luvulla kehitetty tekoälyohjelma MYCIN kykeni potilastietojen perusteella suosittelemaan oikeaa antibioottia potilaalle perustuen potilaan bakteeripatogeeneihin sekä hänen painoonsa. Vastaavia tekoälyjä, jotka auttoivat muun muassa sairauksien diagnosoinnissa on ollut käytössä useita MYCIN jälkeen (Kaul, Enslin ja Gross 2020). Tässä luvussa kerrotaan tekoälyn historiasta, miksi shakkia käytetään niiden testaamiseen sekä puhutaan kahdesta eri tekoälystä ja niiden toteutuksesta.

2.1 Shakki osana tekoälyn kehitystä

Shakki on ajatustasolla ja säännöiltään selkeää, mutta toiminnaltaan hyvin monimutkainen ja täynnä erilaisia mutkikkaita siirtoja ja ratkaisuja. Pelin tarkoituksena on ajateltava pidemmälle kuin vastustaja pelinappuloiden siirron suhteen, ja saavuttaa ylilyöntiasema. Mitä pidemmälle pelaaja ajattelee pelitilannetta ja mahdollisia skenaarioita laudalla, sitä suuremmat mahdollisuudet hänellä on voittaa peli. Juuri sen takia, että kyseisessä pelissä pitää ajatella siirtoja pitkälle tulevaisuuteen, tekee siitä erinomaisen pelikohteen tekoälyille. Sen avulla voidaan arvioida, kuinka tehokkaita kehitettävät tekoälyt ovat.

Shakkia onkin hyödynnetty aiemmin mainitsemani Deep Bluen sekä AlphZeron kehityksessä; ne ovatkin nimenomaan tekoälyohjelmia, jotka on kehitetty pelaamaan shakkia. Koska shakissa on lukematon määrä pelitilanteita ja siirtoja, tekoäly tarvitsee valtavan määrän prosessointivoimaa. Tällä voimalla se pystyy valitsemaan optimaalisimman pelisiirron.

Deep Blue ja AlphaZero ovat hyvin merkittäviä ohjelmia tekoälyn kehityksen historiassa. Vuonna 1997 shakin suurmestari Garry Gasparovin voittanut Deep Blue oli ensimmäinen tietokone, joka päihitti ihmisen virallisessa shakkiottelussa. Yli 50 vuoden kehitystyön jälkeen ihmisen päihittävä shakkia pelaava tietokoneohjelma Deep Blue onnistui siinä viimein vuonna 1997, tehden siitä merkittävän virstapylvään tekoälyn kehityksen historiassa (Newborn ja Newborn 2003). Voitto osoitti, että tietokoneet pystyivät päihittämään ihmisen pe-

leissä, joissa tarvitaan lukuisia monimutkaisia siirtoja ja strategioita.

2.2 Deep Bluen toiminta ja merkitys

Deep Blue on kovakoodattu tietokoneohjelma, joka luotiin vain yhtä tarkoitusta varten: shakin pelaamiseen. Se kehitettiin IBM Research -yhtiön toimesta 1990-luvulla, jolla saatiin luotua maailmanluokan shakkikone vuosien työskentelyn tuloksena. Deep Bluella oli useampia edeltäjiä, joiden kehittäminen johti lopulta kyseessä olevaan viimeisimpään versioon Deep Bluesta. Deep Bluella oli aiempi versio, joka pelasi ja hävisi shakkiottelun shakin suurmestari Garry Kasparovia vastaan vuonna 1996. Vuotta myöhemmin paranneltu versio sai mahdollisuuden uusintaotteluun, jonka lopputuloksena tekoäly voitti Garry Kasparovin ja sai sen myötä paljon huomiota maailmalla (Campbell, Hoane Jr ja Hsu 2002). Tässä tutkielmassa puhutaan vuonna 1997 Kasparovin voittaneesta tekoälystä.

Vaikka monet ovat luonnehtineet kyseisten shakkipelien olleen ihminen vastaan kone, Hsu (2002) kertoo kirjassaan oman näkemyksensä shakkipelistä olleen ennemminkin kaksi ihmistä vastakkain, joilla kummallakin oli eri roolit: ”esiintyjä” (Kasparov) sekä ”työkaluntekijä” (Deep Bluen kehitystiimi). Vuoden 1996 shakkipelin voitti ”esiintyjä” mutta 1997 vuoden pelin voitti lopulta ”työkaluntekijä”. Vuoden 1996 shakkipelin jälkeen IBM:ssä havaittiin ensimmäisen version Deep Blue -tekoälyn sisältävän useita puutteita. Tämän seurauksena ohjelmaa alettiin kehittämään tekoälyn uusintaottelua varten, kuten Campbell, Hoane Jr ja Hsu (2002) tekstissään mainitsee.

Deep Blue sisältää yksisiruisen hakualgoritmin (joka toimii yhdellä CPU:lla), rinnakkain toimivia valtavia järjestelmiä, tehokkaan hakualgoritmin, monimutkaisen arviointifunktion, jolla ohjelma laski mitkä siirroista olivat järkeviä, sekä valtavan tiedoston sille syötetystä datasta. Valtava määrä dataa sisälsi tietoa lukuisista shakkipeleistä, niiden siirroista sekä pelien lopputuloksista. Deep Bluen ohjelmisto koostui kolmesta tasosta: yksi tietokoneen prosessoreista oli valittu ”mestariksi” ja loput ”työntekijöiksi”. Kyseinen ”mestari” tutki shakin pelipuun ylimpiä tasoja, joista se sitten jakoi lehtiasemia ”työntekijöille”. Nämä ”työntekijät” puolestaan jakoivat työnsä alemmille tasoille, jotka viimeistelivät viimeisten tasojen haut (Campbell, Hoane Jr ja Hsu 2002). Vaikka Deep Blue on pääosin kovakoodattu, se sisältää

tästä huolimatta myös ei-kovakoodattuja elementtejä, kuten aiemmin mainittu puuhaku.

Järjestelmän nopeus vaihteli paljon, riippuen juuri meneillään olevan shakkisiirron monimutkaisuudesta. Taktisissa pelitilanteissa, jossa pitkälle ulottuvia tärkeitä pelinappuloiden siirtoja oli olemassa, Deep Blue kykeni laskemaan 100 miljoonaa eri siirtoa sekunnissa. Rauhallisissa yksinkertaisissa pelitilanteissa siirtoja laskettiin jopa 200 miljoonaa. Kuuluisassa shakkipelissä Kasparovia vastaan, korkein saavutettu laskentateho oli 330 miljoonaa shakkisiirtoa sekunnissa. Näihin nopeuksiin pystyttiin Deep Bluen valtavan prosessointitehon ansiosta (Campbell, Hoane Jr ja Hsu 2002).

2.3 AlphaZeron toiminta ja merkitys

Tässä kappaleessa kerrotaan Neuroverkkoja hyödyntävästä AlphaZerosta, joka julkaistiin vuonna 2017. Kyseinen AlphaZero on tietokoneohjelma, joka on tekoälytutkimusyhtiön DeepMindin kehittämä tekoäly. Toisin kuin Deep Blue, AlphaZero hyödyntää ohjelmassaan neuroverkkoja, minkä takia siitä pystyttiin kehittämään oppiva tekoäly. Kuten tutkielmassa aiemmin mainittiin, oppiva tekoäly ei ole riippuvainen siitä, mitä se ohjelmoidaan tekemään. Se kykenee tarpeen mukaan oppimaan itsenäisesti ympäristöstään, esimerkiksi pelaamalla erilaisia pelejä itseään vastaan (David, Netanyahu ja Wolf 2016).

Ennen AlphaZeroa, Microsoftin tytäryhtiö DeepMind oli kehittänyt aiempia versioita kyseisestä tekoälystä: AlphaGo sekä AlphaGo Zero kuten Holcomb ym. (2018) kirjoittamassa artikkelissa kerrotaan. Vaikka AlphaZeron aiemmat versiot ovat merkityksellisiä syväoppivan tekoälyn kannalta, keskitytään tässä kirjallisuuskatsauksessa kuitenkin vain viimeisimpään AlphaZeroon.

DeepMind, joka omistaa AlphaZeron tutkii oppivaa tekoälyä ja se on erityisesti perehtynyt syväoppimiseen. Juuri syväoppimista on hyödynnetty AlphaZeron toteutuksessa sekä sen aiemmissa versioissa. AlphaZeron tehokkaan algoritmin ansiosta, se pystyy saavuttamaan ”superihmisen” suoriutumisen monissa haastavissa peleissä. Vaikka tekoälylle ei annettu muuta kuin pelin ohjeet ja satunnainen pelitilanne, se päihitti senhetkisen maailmanmestariohjelman shakissa. Shakin lisäksi pelejä joissa se päihitti ihmisen oli Shogi (Japanilainen versio shakista) sekä Go (Silver ym. 2018). AlphaZero on itsessään merkkipaalu tekoälyn kehityk-

sessä, Kuten Tomašev ym. (2022) artikkelissaan kirjoittaa.

Googlen DeepMindin kehittämä AlphaZero on merkittävä ohjelma tekoälyn kehityksessä. Pelaamalla toistuvasti itseään vastaan, se kykeni oppimaan pelaamaan shakkia, go:ta sekä shogia ilman että sille annettiin etukäteen mitään tietoa tai pelistrategioita. AlphaZeron kyvyllä oppia ja kehittyä omasta pelaamisestaan sai sen päihittämään monet ihmisvastustajansa kyseisissä peleissä. Tämä teki siitä hyvin merkittävän tekoälyn sen kehityksen historiassa (Silver ym. 2018). Ymmärtämällä Deep Bluen ja AlphaZeron vaikutusta, voimme paremmin ymmärtää tekoälyteknologioiden vaikutusta tulevaisuuteemme.

3 Kovakoodattu tekoäly

Tässä luvussa perehdytään tarkemmin kovakoodattuun tekoölyyn yleisellä tasolla, tuodaan ilmi sen käyttötarkoituksia ja mitä se yleisesti tarvitsee toimiakseen. Kovakoodatun tekoölyn toimintaa tuodaan esille hyödyntäen kuuluisaa tekoälyohjelmaa Deep Bluea.

3.1 Kovakoodattu tekoäly yleisesti

Kovakoodattu tekoäly on tekoälyä, jonka jokainen valinta ja toiminto on koodattu jo valmiiksi lähdekoodiin. Se on rajoittunut toimimaan sille ohjelmoitujen sääntöjen mukaan, toisinkin ympäristöstään oppiva koneoppiva tekoäly. Kovakoodatulla tekoölyllä on suuri määrä sääntöjä ohjelmassaan, joiden pohjalta se tekee päätökset ja ratkaisut ongelmiin. Se ei kykene eikä sen ole tarkoitus kehittää uusia ratkaisuja tai oppimaan saamastaan datasta (Hayes-Roth 1985). Jos kovakoodatulla tekoölyllä olisi tehtävänä etsiä esimerkiksi oikea tilinumero tietokannasta, se toteuttaa tehtävänsä ja etsii oikean tiedon automaattisesti. Se ei kuitenkaan pysty kehittämään tapaa etsiä vastaavaa tietoa tehokkaammin tai sopeutua muuttuviin tietoihin.

Kun kovakoodattua tekoälyä lähdetään kehittämään, sille suunnitellaan tavanomaisesti vain yksi tarkoitus, kuten juuri shakin pelaaminen (Hassabis 2017). Ohjelmalle syötetään valtava määrä dataa shakin ammattilaisten pelistrategioista ja liikkeistä, joita kyseisen tekoölyn hakualgoritmi käy läpi pelatessaan vastustajaansa vastaan. Koska eri pelitilanteita on valtavasti, tekoäly käy datan läpi raa'an voiman avulla ja etsii parhaan mahdollisen ratkaisun kyseiseen pelitilanteeseen. Tämä tapa vaatii ohjelmalta valtavasti prosessointitehoa sekä tehokkaan hakualgoritmin.

Tekoälyä verrataan usein vain koneeseen, joka oppii itse ympäristöstään ja omasta toiminnastaan. Tämä ei kuitenkaan pidä täysin paikkaansa. Tekoäly voi olla sääntöpohjainen järjestelmä, jossa on lukuisia kovakoodattuja sääntöjä, esimerkiksi ”jos numero on kaksi, tulosta näytölle 'Hei' ”. Tällaisia sääntöjä voi olla järjestelmässä kymmeniäkin tuhansia. Kun kaikki nämä säännöt ketjutetaan yhteen, voi kyseinen järjestelmä tehdä päätöksiä tai suorittaa erilaisia tehtäviä uusilla saamallaan tiedoilla. Tällä tavalla järjestelmä voi toimia itsenäisesti ilman, että ihmisen tarvitsee puuttua sen toimintaa (Namatherdhala, Mazher ja Sriram 2022).

3.2 Kovakoodatun tekoälyn toiminta ja käyttötarkoitus

Kuten kappaleessa 3.1 mainittiin, kovakoodatussa ohjelmassa data on kirjoitettuna valmiiksi ohjelman lähdekoodiin. Tekoälyn ei silloin tarvitse eikä se edes voi saada dataa muualta ympäristöstään. Tämän takia kovakoodattu ohjelma tarvitsee toimiakseen mm. tehokkaan hakualgoritmin sekä paljon prosessointivoimaa, jotta se voisi käydä sille syötetyn datan nopeasti läpi. Näitä ominaisuuksia tarvitaan, jotta ohjelma pystyisi suorittamaan sille annetun tehtävän nopeasti järkevässä ajassa. Esimerkiksi 1990-luvulla kehitetyllä kovakoodatulla tekoälyohjelmalla Deep Bluella oli käytössään 256 prosessoria, jotka toimivat samanaikaisesti rinnakkain ja jotka arvioivat jopa 200 miljoonaa erilaista shakkisiirtoa sekunnissa (“Deep Blue” 2004).

Kuten kappaleessa 2.1 mainitsin, kovakoodattua tekoälyä hyödynnetään monissa eri paikoissa. Näitä ovat erilaiset yksinkertaiset asiakaspalveluchatit, äänen ja kuvantunnistusohjelmat, sekä esimerkiksi tuotannossa koneiden hallinnassa ja laadunvarmistuksessa. Nämä kaikki hyödyntämiskohteet toimivat hyvin kovakoodatulla tekoälyllä, sillä ne ovat toistoa vaativia tehtäviä. Vaikka ohjelmat pystyisivätkin kasvattamaan osaamistaan niiden tietokantoja suurentamalla, ne ovat suunniteltu ratkaisemaan pääasiassa jo tunnettuja ongelmia niille ohjelmoitujen sääntöjen mukaan (Hayes-Roth 1985).

Datan ollessa valmiiksi syötettynä kovakoodatulla ohjelmalla, se ei voi oppia suorittamaan uusia tehtäviä eikä sopeutumaan muutoksiin. Tämä voi olla ongelmana, jos tehtävänä on suorittaa kyseiselle tekoälylle tuntematon tehtävä. Monissa asiakaspalveluchateissa tulee usein vastaus monimutkaisiin kysymyksiin: ”En ymmärrä kysymystäsi”. Kyseiselle tekoälyohjelmalle on annettu vastausmahdollisuudet vain tiettyihin kysymyksiin, jolloin vastaavia ”en osaa vastata” -tilanteita voi tulla usein. Vaikka kovakoodattu tekoäly ei pystykään oppimaan toiminnastaan tai ympäristöstään, sitä voidaan silti kehittää paremmaksi. Kovakoodatun tekoälyn suorituskykyä voidaan parantaa esimerkiksi kehittämällä sen hyödyntämiä hakualgoritmeja, lisäämällä enemmän dataa ohjelmalle hyödynnettäväksi tai yksinkertaisesti parantamalla sen tietokonejärjestelmiä (Hayes-Roth 1985).

4 Neuroverkkoihin pohjautuva tekoäly

Tässä luvussa kerrotaan neuroverkkoja hyödyntävästä tekoälystä yleisellä tasolla ja selitetään hieman niiden toiminnasta. Kappaleessa 4.2 puhutaan tarkemmin neuroverkkojen syötteistä, painoista ja aktivaatiofunktioista. Viimeisessä kappaleessa kerrotaan lopuksi vielä neuroverkkojen käyttötarkoituksesta.

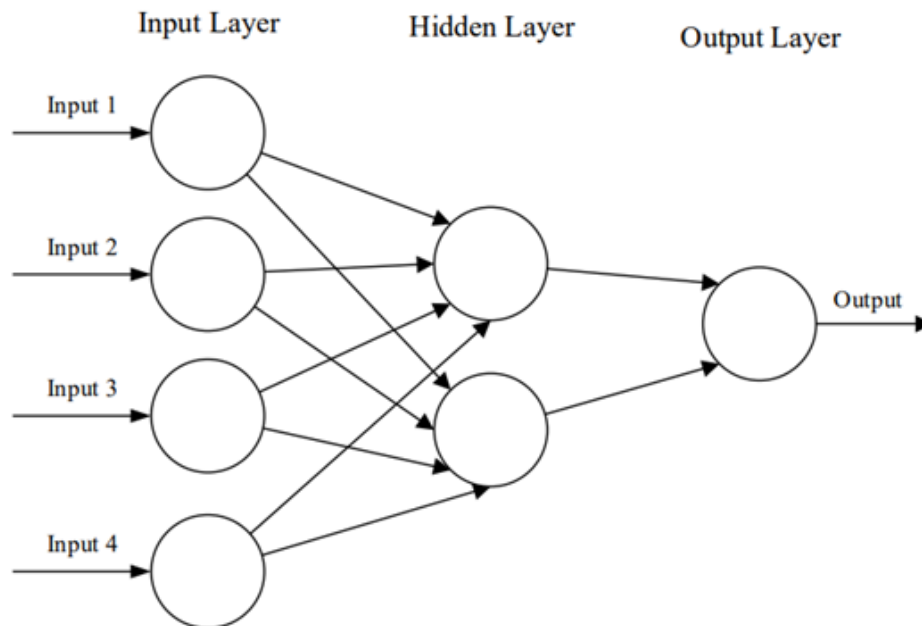
4.1 Neuroverkot yleisesti

Keinotekoiset neuroverkot ovat yksi koneoppivan tekoälyn alalajeista, kuten O’Shea ja Nash (2015) artikkelissaan mainitsee. Neuroverkot ovat laskennallisia järjestelmiä, joihin on otettu vahvasti mallia oikeista biologisista hermostojärjestelmistä. Näitä ovat esimerkiksi ihmisaiivot. Neuroverkot koostuvat suurilta osin valtavasta määrästä toisiinsa yhteyksissä olevista laskentayksiköistä, joita kutsutaan neuroneiksi. Nämä neuronit toimivat ja oppivat yhteistyössä, optimoidakseen saamansa lopullisen tuloksen (O’Shea ja Nash 2015).

Neuroverkot ovat hyvin tehokkaita ratkaisemaan erilaisia ongelmia sekä tulkitsemaan dataa, mitä voidaan kiittää niiden kyvystä oppia niille annetusta informaatiosta (Krogh 2008). Neuroverkoissa on tasoja, joissa neuronit sijaitsevat. Tasojen neuronit aktivoituvat riippuen niiden saamista signaaleista, jotka lähettävät sitten vuorostaan signaaleja eteenpäin itseensä kytkettyihin neuroneihin. Yhdessä ne muodostavat suuren verkon, missä jokainen neuron on yhteydessä toiseen jotakin reittiä pitkin. Yksinkertaisessa neuroverkossa näitä tasoja on vain muutama, mutta syväoppivassa neuroverkossa tasoja voi olla hyvinkin suuri määrä. Mitä enemmän tasoja neuroverkoilla on, sitä monimutkaisempia ja syvempiä tehtäviä se kykenee ratkaisemaan. Ensimmäiset kerrokset voivat tulkita esimerkiksi reunoja tai kulmia, mikäli syötetty data on otettu kuvasta. Myöhemmät kerrokset voivat tämän jälkeen yhdistellä muotoja muodostaen siitä kokonaisen kuvan. Aina ei välttämättä edes tiedetä, mitä kaikkien kerroksien välillä tapahtuu (Schmidhuber 2015).

Neuroverkkojen rakennetta voidaan mallintaa yksinkertaisella kuvalla kuvassa 1.

Data joka halutaan syöttää neuroverkolle ladataan yleensä syötekerrokseen (*input layer*) mo-



Kuvio 1. Yksinkertainen kolmitasoinenneuroverkko (O’Shea ja Nash 2015, sivu 2) Kuvassa on mallinnettuna syötekerros (*input layer*), piilokerros (*hidden layer*) sekä tulostuskerros (*Output layer*), jotka ovat tyypilliset lukuisille yleisille eri neuroverkko arkkitehtuureille.

niulotteisena vektorina, mikä vuorostaan jakaa datan piilokerrokseen (*hidden layers*). Piilokerrokset tekevät päätöksiä pohjautuen edelliseen kerrokseen ja arvioivat, miten stokastinen muutos niissä heikentää tai vahvistaa lopputulosta (*Output*). Neuroverkkojen toteuttamaa lopputuloksen arvioimista kutsutaan oppimiseksi. Kun neuroverkoissa on lukuisia päällekkäin pinottuja piilokerroksia, sitä kutsutaan silloin yleisesti syväoppimiseksi (O’Shea ja Nash 2015).

4.2 Syötteet, painot ja aktivaatiofunktiot

Kuten kappaleessa 3.1 mainitaan, tyypillisessä neuroverkossa on kolme kerrosta. Syötekerros, piilokerros sekä tulostuskerros. Syötteet, mitä neuroverkoille voidaan antaa voi olla vektorien lisäksi kuvia, tekstiä tai muita numeerisia arvoja. Ne tulee tästä huolimatta muuttaa ensin dataksi, jota neuroverkot pystyvät tulkitsemaan. Eri neuroverkot voivat hyödyntää näitä riippuen niiden objektiivista ja sen hetkisetä toimintatarkoituksesta (O’Shea ja Nash 2015).

Käytettävän neuroverkon on oltava konfiguroituna ja ohjelmoituna niin, että joukko syötteitä tuottavat halutut lopputulokset (Abraham 2005).

Kappaleessa 3.1 mainittujen neuronien välillä on myös painoja (*weights*), joiden perusteella neuroverkko määrittää, mitä reittiä tieto kulkee neuroverkkojen eri kerrosten välillä. Jos verkko on suuri ja sillä on sopiva rakenne sekä oikein valitut painot, se voi teoriassa suorittaa hyvinkin haastavia funktioita tietyt rajoitukset täyttäen (Dongare, Kharde, Kachare ym. 2012). Kun painoja muokataan neuronien välillä, kutsutaan sitä oppimiseksi. Se on prosessi, jossa neuroverkon satunnaisarvoiset parametrit muokkautuvat jatkuvasti ympäristönsään suoritettavien simulaatioiden perusteella (Dongare, Kharde, Kachare ym. 2012).

Neuroverkoissa on neuronien sekä painojen lisäksi aktivaatiofunktioita (*activation functions*), jotka määrittävät, miten data käsitellään ja millaisia aktivaatioita tuotetaan kussakin neuronissa. Siinä missä painot vaikuttavat neuronien välillä tiedon kulkureittiin neuronien välillä, aktivaatiofunktiot määräävät miten data käsitellään ja millaisia aktivaatioita kussakin neuronissa tuotetaan. Aktivaatiofunktiot ovat tärkeitä neuroverkoissa, sillä ne mahdollistavat ei-lineaarisen oppimisen monimutkaisesta datasta tai -suhteista tietoaaineistoissa. Se mahdollistaa tehokkaan ja tarkan mallintamisen, jos tarkoituksena on esimerkiksi tunnistaa kuvia. Ilman aktivaatiofunktioita, tuloste olisi neuroverkoissa yksinkertainen. Kyseinen neuroverkko ei kykenisi oppimaan tai tunnistamaan kuvaajia tai sille syötettyä tietoa (Sharma, Sharma ja Athaiya 2017).

4.3 Toiminta ja käyttötarkoitus

Neuroverkkoja pystytään hyödyntämään lukuisissa eri kohteissa. Kuten Dreyfus (2005) toteaa kirjassaan, neuroverkot voivat olla hyödyllisiä ongelmissa joissa tarvitaan etsimistä, koneoppimista tai ei-lineaarista suhdetta numeroiden välillä. ei-lineaarisella tarkoitetaan tässä esimerkiksi biologisia prosesseja tai taloudellisia ilmiöitä, joiden tutkimiseen tarvitaan tehokasta tekoälyä. Tämän takia täytyy ensimmäisenä selvittää, tarvitaanko ongelmanratkaisussa juuri neuroverkkoja.

Yksi kriteeri neuroverkkojen käytölle on sen tarpeellisuus. Jos ongelman pystyy ratkaisemaan tavallisella lineaarisella mallilla, silloin ei kannata tuhjata resursseja neuroverkon luo-

miseen ja käyttämiseen. Jos sopivia tietojoukkoja on saatavilla, neuroverkkoja voidaan käyttää hyvin monissa ongelmanratkaisua vaativissa tehtävissä. Data tulee muuntaa neuroverkon ymmärtämään muotoon silloin, jos data ei ole numeerista vaan esimerkiksi symbolista (Dreyfus 2005).

Koska neuroverkot ovat erinomaisia ratkaisemaan ongelmia joissa on ei-lineaarisia suhteita Dreyfus (2005), niitä voidaan käyttää lukuisissa eri käyttötarkoituksissa. Näitä ovat esimerkiksi kuvan- ja puheentunnistus, luonnollisen kielen käsittely, lääketieteellinen diagnosointi kuten luvussa 2 mainittiin, autonomiset ajoneuvot sekä erilaiset taloudelliset ennusteet. Nämä kaikki ovat käyttökohteita, joiden ongelmien ratkaisemiseen voidaan käyttää neuroverkkoja.

5 Tekoälyjen vertailu

Tässä luvussa keskitytään vertailemaan aiemmissa luvuissa ilmi tulleiden tekoälyjen eroja sekä niiden asemaa tulevaisuudessa. Tämän lisäksi mainitaan lyhyesti tekoälystä nimeltä ChatGPT ja pohditaan sen vaikutuksia opetuksen sekä turvallisuuden kysymyksiin.

5.1 Tekoälyjen vaikutus tulevaisuudessa

Luvussa 2 mainittiin, että kovakoodattua tekoälyä käytetään toistoa vaativissa yksinkertaisissa tehtävissä. Näitä ovat esimerkiksi asiakaspalveluchatit ja tuotantoyksiköissä koneiden hallinta. Koska tämän tekoälyn ei tule oppia ympäristöstään ja tehtävistään, on se suunniteltu toimimaan aina samalla tavalla poistaen siltä ominaisuuden kehittyä ja sopeutua uusiin ympäristöihin. Tämä rajoittaa kovakoodattua tekoälyä paljon, jos vertailtavana kohteena on tekoäly joka toimii neuroverkoilla. Sen kyky sopeutua erilaisiin tehtäviin ja uusien asioiden oppiminen tekevät siitä ylivertaisen tekoälymuodon.

Vaikka kovakoodattu tekoäly ei ole yhtä joustava kuin oppiva tekoäly, kovakoodattua tekoälyä voidaan edelleen käyttää tulevaisuudessa useissa yksinkertaisissa sekä toistuvissa tehtävissä. Näitä ovat esimerkiksi teollisuuden monet alat sekä erilaiset äänen- ja kuvantunnistus ohjelmat, joista mainitaan luvuissa 3 ja 4. Vaikka neuroverkot oppivat suorittamistaan tehtävistä, myös kovakoodattua tekoälyä voidaan kehittää parantamalla sen hakualgoritmeja tai päivittämällä järjestelmää. Näitä molempia tekoälyjä voitaisiin mahdollisesti hyödyntää yhdessä samassa järjestelmässä. Kun molempien menetelmien parhaat puolet yhdistää, voitaisiin saada kehitettyä entistä kehittyneempi järjestelmä.

Neuroverkot tulevat luultavasti silti olemaan suurempi osa tulevaisuuden tekoälyä. Vielä on silti hyvin vaikeaa sanoa, millaisen roolin neuroverkot ottavat ihmiskunnan tulevaisuudessa, kuten Nielsen (2015) kirjassaan mainitsee. Niiden kyvyllä oppia ja sopeutua muuttuviin ympäristöihinsä tekevät niistä monikäyttöisempiä sekä laaja-alaisempia kuin kovakoodattu tekoäly. Vaikka kovakoodattuihin ohjelmiin perustuva tekoäly tulee luultavasti väistymään neuroverkkojen tieltä, on sillä silti ollut merkittävä rooli tekoälyn kehityksen historiassa. Ilman Deep Bluen kaltaisten ohjelmien aikaansaamia saavutuksia, nykyiset AlphaZeron tyypp-

piset oppivat tekoälyt eivät välttämättä olisi samanlaisia sekä yhtä kehittyneitä kuin mitä ne tänä päivänä ovat.

5.2 ChatGPT

Joulukuussa 2022 yritys nimeltä OpenAI julkaisi uuden tekoälyohjelman nimeltä ChatGPT. Tämä tekoälyohjelma kykenee tulkitsemaan ihmisten kysymyksiä sekä vastaamaan niihin luontevasti monella eri kielellä. Tämä onnistuu koneoppimisella sekä luonnollisen kielen käsittelyllä, jota ChatGPT käyttää tehden siitä hyvin vuorovaikutteisen chattitekoälyn (Thorp 2023).

ChatGPT:n kirjoitusmahdollisuudet ovat todella laajat. Ohjelma pystyy esimerkiksi kirjoittamaan esseevastauksia, tieteellisiä artikkeleita tai jopa elokuvakäsikirjoituksia. Näiden tekstien laadussa voi tosin olla vielä parantamisen varaa, sillä ChatGPT viittaa kirjoituksissaan joskus myös olemattomiin artikkeleihin tai lähteisiin. Toisinaan ne antavat vastauksissaan täysin epävirallista tietoa (Thorp 2023).

ChatGPT:n kohdalla on ollut useita huolenaiheita, kuten esimerkiksi opiskelijoiden oppiminen. Vaikka kyseinen tekoäly onkin hyvä työkalu oppimisen, tiedonhaun ja lukemisen tueksi, sitä voidaan käyttää myös väärin. Jos opiskelijat käyttävät kyseistä tekoälyä esseidensä kirjoittamiseen tai tehtäviensä tekemiseen, he eivät kehitä omia taitojaan niissä. Tekoäly on hyödyllistä, jos opettajat ja opiskelijat voivat hyödyntää sitä opetuksen tukena. Vielä alkuvaiheessa ihmisten on osattava hahmottaa kyseisen tekoälyn ongelmat, ja sen potentiaalia mm. oppimisen tukemisessa on arvioitava kriittisesti (Kasneci ym. 2023). Yksi ongelma vielä suuremmissa mittakaavassa on luotettavien tieteellisten artikkelien, sekä tekoälyn kirjoittamien artikkelien sekoittuminen toisiinsa. Tämä on yksi syistä, miksi useat maat ovat jo estäneet ChatGPT:n käytön kokonaan.

6 Yhteenveto

Tässä tutkielmassa oli tavoitteena kertoa kahdesta tekoälyn kehittämiskäytännöstä, jotka olivat kovakoodattu tekoäly, sekä koneoppivan tekoälyn alaluokkana olevat neuroverkot. Tutkielmassa käytiin läpi kovakoodatun tekoälyn rakennetta, tekoälyn kehityksen historiaa sekä hetkeä, missä tekoäly päihitti ihmisen ensimmäistä kertaa ihmiskunnan historiassa. Vuoden 1997 shakkipeli Deep Bluen ja sen hetkisen shakin maailmanmestarin välillä asetti historiallisesti merkittävän virstapylvään tekoälyn kehityksessä.

Tutkielmassa selvitettiin kovakoodatun tekoälyn lisäksi uudempaa menetelmää kehittäviä tekoälyjä, jotka olivat neuroverkot. Tutkielmassa havaittiin, että neuroverkkojen toimintaperiaate, ominaisuudet sekä tulevaisuuden hyödyntämismahdollisuudet eri aloilla takaavat niille mahdollisuuden olla merkittävä osa ihmisten tulevaisuutta. Niiden kyky oppia sopeutumaan moniin eri käyttötarkoituksiin vaikuttaa suuresti tähän mahdollisuuteen.

Deep Bluen voitto kuuluisaa shakkimestaria vastaan merkitsi tekoälyn virstapylvään, mikä herätti kiinnostusta tekoälyn soveltamisesta monella eri alalla. Tekoälyä hyödynnetään tänäkin päivänä peliteknologiassa, tiedonkäsittelyssä sekä lääketieteessä. Vaikka AlphaZero onkin uudempi ohjelmisto, sen ominaisuudet ja sopeutumiskyvyt päihittivät Deep Bluen ohjelmiston. Tätä voitaisiin pitää yhtenä perusteluna, miksi neuroverkkoja tullaan hyödyntämään tulevaisuudessa enemmän kuin kovakoodattua tekoälyä.

Tutkittavaa materiaalia ollessa paljon tarjolla, sen läpikäynti kriittisesti tarkastellen oli haastavaa. Valtavan tiedonmäärän takia aihetta oli rajattava niin, että tutkielmasta olisi saatu looginen sekä ymmärrettävä kokonaisuus. Kyseisen tutkielman innoittamana neuroverkkojen tulevaisuuden käyttötarkoituksia voitaisiin selvittää vielä tarkemmin. Neuroverkkoja ollessa useita erilaisia, voitaisiin myös niitä lähteä tutkimaan erikseen ja selvittämään esimerkiksi niiden tarkempaa toimitapaa sekä kehitystä.

Lähteet

- Abraham, Ajith. 2005. "Artificial neural networks". *Handbook of measuring system design*.
- Campbell, Murray, A Joseph Hoane Jr ja Feng-hsiung Hsu. 2002. "Deep blue". *Artificial intelligence* 134 (1-2): 57–83.
- David, Omid E, Nathan S Netanyahu ja Lior Wolf. 2016. "Deepchess: End-to-end deep neural network for automatic learning in chess". Teoksessa *Artificial Neural Networks and Machine Learning–ICANN 2016: 25th International Conference on Artificial Neural Networks, Barcelona, Spain, September 6-9, 2016, Proceedings, Part II 25*, 88–96. Springer.
- "Deep Blue". 2004. Viitattu 27. huhtikuuta 2023. <https://www.britannica.com/topic/Deep-Blue>.
- Dongare, AD, RR Kharde, Amit D Kachare ym. 2012. "Introduction to artificial neural network". *International Journal of Engineering and Innovative Technology (IJEIT)* 2 (1): 189–194.
- Dreyfus, Gérard. 2005. *Neural networks: methodology and applications*. Springer Science & Business Media.
- Floridi, Luciano. 2019. "What the near future of artificial intelligence could be". *Philosophy & Technology* 32:1–15.
- Hassabis, Demis. 2017. *Artificial intelligence: chess match of the century*.
- Hayes-Roth, Frederick. 1985. "Rule-based systems". *Communications of the ACM* 28 (9): 921–932.
- Holcomb, Sean D., William K. Porter, Shaun V. Ault, Guifen Mao ja Jin Wang. 2018. "Overview on DeepMind and Its AlphaGo Zero AI", 67–71. <https://doi.org/10.1145/3206157.3206174>.
- Hsu, Feng-Hsiung. 2002. *Behind Deep Blue: Building the computer that defeated the world chess champion*. Princeton University Press.

Kasneji, Enkelejda, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günemann, Eyke Hüllermeier ym. 2023. “ChatGPT for good? On opportunities and challenges of large language models for education”. *Learning and Individual Differences* 103:102274.

Kaul, Vivek, Sarah Enslin ja Seth A Gross. 2020. “History of artificial intelligence in medicine”. *Gastrointestinal endoscopy* 92 (4): 807–812.

Klein, Dominik. 2022. “Neural Networks for Chess”. *arXiv preprint arXiv:2209.01506*.

Krogh, Anders. 2008. “What are artificial neural networks?” *Nature biotechnology* 26 (2): 195–197.

Namatherdhala, Bharatwaja, Noman Mazher ja Gopal Krishna Sriram. 2022. “Artificial Intelligence in Product Management: Systematic review”. *International Research Journal of Modernization in Engineering Technology and Science* 4 (7).

Newborn, Monroe, ja Monty Newborn. 2003. *Deep Blue: an artificial intelligence milestone*. Springer Science & Business Media.

Nielsen, Michael A. 2015. *Neural networks and deep learning*. Nide 25. Determination press San Francisco, CA, USA.

O’Shea, Keiron, ja Ryan Nash. 2015. “An introduction to convolutional neural networks”. *arXiv preprint arXiv:1511.08458*.

Poola, Indrasen. 2017. “How artificial intelligence in impacting real life everyday”. *International Journal for Advance Research and Development* 2 (10): 96–100.

Schmidhuber, Jürgen. 2015. “Deep learning in neural networks: An overview”. *Neural Networks* 61:85–117. ISSN: 0893-6080. <https://doi.org/https://doi.org/10.1016/j.neunet.2014.09.003>. <https://www.sciencedirect.com/science/article/pii/S0893608014002135>.

Sharma, Sagar, Simone Sharma ja Anidhya Athaiya. 2017. “Activation functions in neural networks”. *Towards Data Sci* 6 (12): 310–316.

Silver, David, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel ym. 2018. “A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play”. *Science* 362 (6419): 1140–1144.

Thorp, H Holden. 2023. *ChatGPT is fun, but not an author*, 6630.

Tomašev, Nenad, Ulrich Paquet, Demis Hassabis ja Vladimir Kramnik. 2022. “The Chess Metagame: Machine Learning as a Playground for Generative Models”. *Communications of the ACM* 65 (2): 60–66. <https://doi.org/10.1145/3460349>.