Niko Klemetti

Koneoppimisen soveltaminen sulautetussa ympäristössä

Insinööri (AMK)

Tieto- ja viestintätekniikan koulutus

Kevät 2023

Tiivistelmä

Tekijä(t): Klemetti Niko

Työn nimi: Koneoppimisen soveltaminen sulautetussa ympäristössä

Tutkintonimike: Insinööri (AMK), tieto- ja viestintätekniikka

Asiasanat: TinyML, IoT, Sulautettu tekoäly.

Opinnäytetyön tavotteena oli tutkia, miten koneoppimista voidaan toteuttaa sulautetussa ympäristössä. Opinnäytetyö aloitettiin tutustumalla tekoälyn historiaan 1900- ja 2000-luvulla. Sen jälkeen perehdyttiin datatieteeseen ja sen osa-alueisiin, tekoälyyn, koneoppimiseen ja syväoppimiseen.

Tekoälyn historian ja datatieteiden yleiskatsauksen jälkeen painopiste siirrettin koneoppimiseen ja sen toteutukseen sulauteetussa ympäristössä. Lisäksi puhuttiin sulautetusta tekoälystä ja tutkittiin takoälymallin optimointitapoja. Tämän jälkeen syvennyttiin TinyML:ään, yhteen sulautetun tekoälyn osa-alueeseen.

Työ huipentui esimerkkisovellukseen, jonka tarkoituksena oli harjoitella tietojen keräämistä, neuroverkon koulutusta sekä reaaliaikaista testausta. Lopputuloksena syntyi ääniohjattu kaukosäädin, joka kykeni vastaamaan äänikomentoihin.

KAMK • University of Applied Sciences

Abstract

Author(s): Klemetti Niko

Title of the Publication: Utilizing machine learning in an embedded environment

Degree Title: Bachelor of Engineering, Information and Communication Technology

Keywords: TinyML, IoT, Embedded AI.

The goal of this thesis was to investigate how machine learning can be implemented in an embedded environment. The thesis began by getting to know the history of artificial intelligence in the 20th and 21st centuries. This was followed by getting familiar ourselves with data science and its sub-areas, artificial intelligence, machine learning and deep learning.

After an overview of the history of artificial intelligence and data science, the focus shifted to machine learning and its implementation in an embedded environment. In addition, embedded artificial intelligence was discussed, and ways of optimizing the artificial intelligence model was studied. After that, the work delved into TinyML, one of the areas of embedded artificial intelligence.

The work culminated in an example application the purpose of which was to practice data collection, neural network training, and real-time testing. The end result was a voice-controlled remote that could respond to voice commands.

Alkusanat

Tämä opinnäytetyö on tehty Kajaanin ammattikorkeakoululle.

Haluan kiittää oppilaitoksen edustajaa ja työn ohjaajana toiminutta Asko Kinnusta neuvoista, työn ohjauksesta ja valvonnasta.

Sisällys

1 Johdanto

Tämän opinnäytetyön tavoitteena on tutkia, miten koneoppimista voidaan soveltaa sulautetussa ympäristössä. Tämän työn avulla pyritään saamaan syvempi ymmärrys sulautetusta tekoälystä ja sen käyttötarkoituksista. Työn toimeksiantaja on Kajaanin ammattikorkeakoulu.

Sulautettu tekoäly on suhteellisen uusi tekoälyn osa-alue, mikä tekee työstä haastavan faktapohjaisen ja tieteellisesti korrektin materiaalin etsimisen suhteen.

Työ aloitetaan yleiskatsauksella tekoälyn historiasta 1900- ja 2000-luvulla. Sen jälkeen perehdytään datatieteeseen ja sen osa-alueisiin, tekoälyyn, koneoppimiseen ja syväoppimiseen. Tämän jälkeen syvennytään sulautettuun tekoälyyn ja tutkitaan mahdollisia tekoälymallin optimointitapoja. Tämän jälkeen puhutaan TinyML:stä, joka on yksi sulautetun tekoälyn osa-alueista. Lopuksi tehdään esimerkkiharjoitus, joka liittyy sulautettuun tekoälyyn.

Työn tavoitteena on oppia tekoälystä sekä kehittää omaa tiedonhankinta- ja kirjoitustaitoa.

2 Tekoälyn historia

2.1 Tekoälyn ensiaskeleet

Vuonna 1943 Warren McCulloch ja Walter Pits tekivät ensimmäisen tutkimuksen ilmiöstä, joka nykyään tunnetaan tekoälynä. He ehdottivat mallia keinotekoisista hermosoluista. [1.]

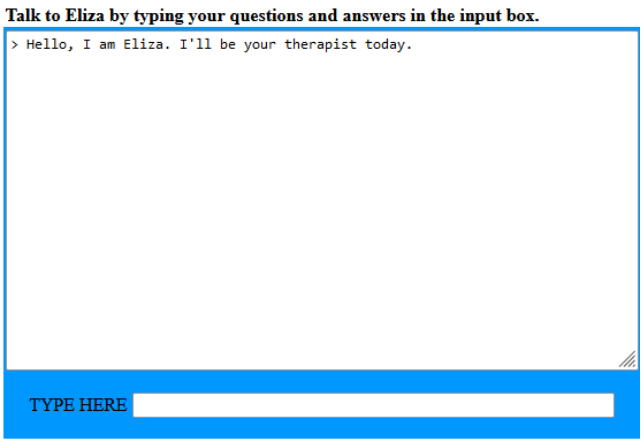
Vuonna 1949 Donald Hebb teorisoi kirjassaan “Organization of Behavior” päivityssäännön hermosolujen välisen yhteyden voimakkuuden muuttamiseksi. Hänen teoriaansa kutsutaan nykyään hebbian-oppimiseksi. [2.]

Vuonna 1950 Alan Turing, joka oli englantilainen matemaatikko ja koneoppimisen pioneeri, julkaisi “Computing Machinery and Intelligence” -kirjan, jossa hän ehdotti koetta, jolla voitaisiin testata koneen kykyä osoittaa älykästä, ihmisen kaltaista, käyttäytymistä. Tätä koetta kutsutaan Turingin testiksi. [3.]

Vuonna 1955 Allen Newell ja Herbert A. Simon loivat ensimmäisen tekoälyohjelman, joka tunnettiin nimellä “Logiikkateoreetikko”. Tämä ohjelma todisti 38 matemaattista lausetta “Principia Mathematican”- kirjan 52 lauseesta. [4.]

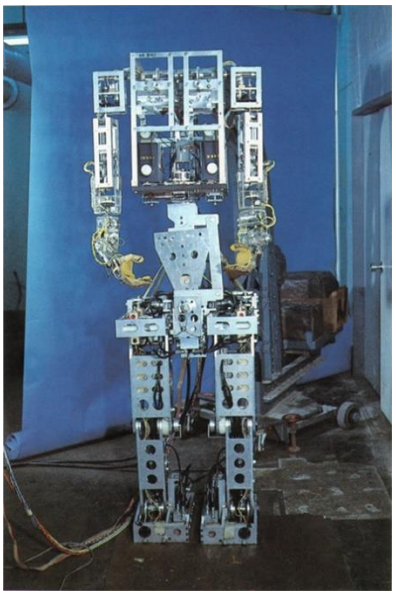
Vuonna 1956 sana “Artificial Intelligence” otettiin ensimmäisen kerran käyttöön amerikkalainen tietotekniikan tutkijan John McCarthyn toimesta Dartmouthin konferenssissa, jonka seurauksena tekoälystä tuli akateeminen ala. [5.]

Vuosina 1964–1966 MIT:n tietojenkäsittelytieteilijä Joseph Weizenbaum loi ensimmäisen chatbotin, joka nimettiin ELIZA:ksi. ELIZA kykeni kommunikoimaan englanniksi ja jopa simuloimaan psykoterapeutin dialogeja. [6.] Alla kuvassa 1 on kuvankaappaus chatbot Elizasta.



Kuva 1. Chatbot Eliza

Vuonna 1972 Japanissa rakennettiin ensimmäinen älykäs humanoidirobotti, joka sai nimekseen WABOT-1. WABOT-1 pystyi kommunikoimaan henkilön kanssa japaniksi, mittaamaan etäisyyksiä ja suuntaa esineisiin käyttämällä ulkoisia reseptoreita, tekokorvia ja -silmiä sekä tekosuuta. WABOT-1 käveli alaraajoillaan ja pystyi tarttumaan ja kuljettamaan esineitä käsillään, jotka käyttivät tuntoantureita. [7.] Alla kuvassa 2 on kuva WABOT-1:sta.



Kuva 2. WABOT-1 [8].

Vuosien 1974 ja 1980 välistä aikakautta kutsutaan ensimmäiseksi tekoälytalveksi. Tekoälytalvella tarkoitetaan ajanjaksoa, jolloin tietojenkäsittelytieteilijät kärsivät vakavasta tekoälyn tutkimukseen liittyvästä valtion rahoituksen puutteesta. Tekoälytalvien aikana kiinnostus tekoälyyn väheni.

Vuonna 1980 tietotekniikan tutkija Edward Feigenbaum kehitti tietokoneen, joka kykeni imitoimaan ihmisen tekemiä päätöksiä. Tietokone käytti hyväkseen asiantuntijajärjestelmää, joka käytti tekoälymenetelmiä ratkaistakseen ongelmia, jotka tavallisesti vaatisivat ihmisen asiantuntemusta. [9.] Samana vuonna Stanfordin yliopistossa pidettiin American Association of Artificial Intelligence -järjestön ensimmäinen kansallinen konferenssi. [10.]

Vuosien 1987 ja 1993 välistä aikakautta kutsutaan toiseksi tekoälytalveksi. Sijoittajat ja Yhdysvaltojen hallitus vähensivät tekoälytutkimuksen rahoituksen korkeiden kustannusten ja heikkojen tulosten vuoksi. Kuitenkin jotkut järjestelmät, kuten XCON, olivat erittäin kustannustehokkaita. [11.]

Vuonna 1997 IBM:n rakentama Deep Blue-supertietokone voitti shakin maailmanmestarin Gary Kasparovin. Siitä tuli ensimmäinen tietokone, joka voitti shakin maailmanmestarin. [12.]

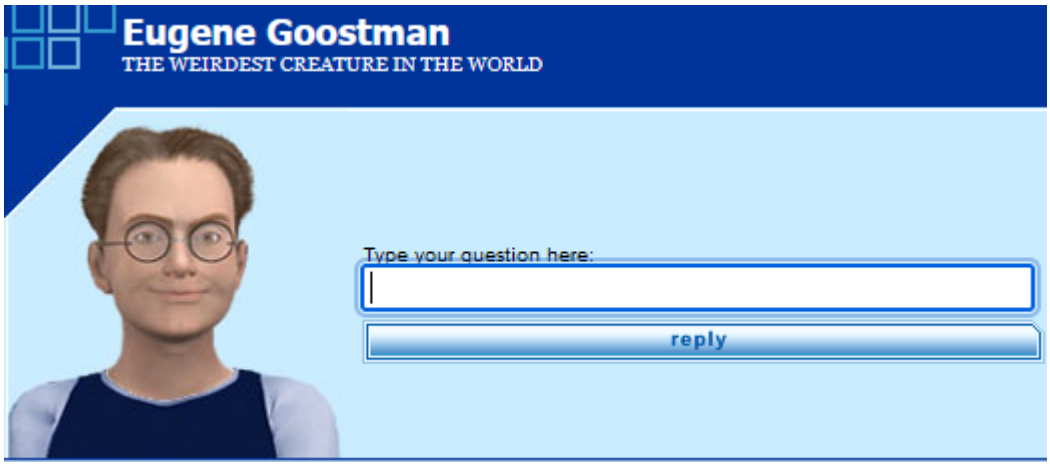
2.2 Tekoäly 2000-luvulla

Vuonna 2002 tekoäly tuli ensimmäistä kertaa ihmisten koteihin Roomban, älykkään pölynimurin, muodossa. Siinä oli antureita, joiden avulla imuri kykeni navigoimaan kodin lattialla. Lisäksi se kykeni havaitsemaan esteitä, likaisia kohtia ja jyrkkiä pudotuksia. [13.]

Vuonna 2011 IBM:n Watson-tietokone voitti tietokilpailun, jossa sen piti ratkaista monimutkaisia kysymyksiä ja arvoituksia. Watson oli osoittanut, että se ymmärtää luonnollista kieltä ja pystyy ratkaisemaan vaikeita kysymyksiä nopeasti. [14.]

Vuonna 2012 Google julkaisi Android-sovellusominaisuuden “Google Now”, joka pystyi seuraamaan käyttäjien hakuhistoriaa, kalentereita ja muita tietoja pysyäkseen ajan tasalla siitä, mitä he saattavat haluta. Käytännössä Google Nowilla varustettu laite oppii asioita käyttäjästään voidakseen tarjota asiaankuuluvampaa tietoa. [15.]

Vuonna 2014 chatbot nimeltä “Eugene Goostman” voitti kilpailun kuuluisassa “Turing-testissä”. Se onnistui vakuuttamaan 1/3 tuomareista siitä, että se on oikea ihminen viiden minuutin keskustelun perusteella. [16.] Alla kuvassa 3 on kuvankaappaus chatbot Eugene Goostmanista.



Kuva 3. Chatbot Eugene Goostman

Vuonna 2018 IBM:n “Project Debater” keskusteli monimutkaisista aiheista kahden mestarikeskustelijan kanssa ja suoriutui siitä erittäin hyvin. Project Debater käsittelee valtavan määrän tekstejä, rakentaa hyvin jäsennellyn puheen tietystä aiheesta, esittää sen selkeästi ja tarkoituksenmukaisesti sekä pyrkii kumoamaan vastustajansa. Lopulta Project Debater auttaa ihmisiä järkeilemään tarjoamalla vakuuttavia, tieteelliseen dataan perustuvia argumentteja. [17.]

3 Tekoäly, koneoppiminen ja syväoppiminen

3.1 Tekoäly

Tekoäly on ihmisen älykkyyden simulointia koneiden avulla. Tekoälyllä on joitakin ihmisen älyn kaltaisia ominaisuuksia, kuten suunnittelu, ongelmien ratkaisu, tiedon esittäminen, liike sekä oppiminen. [18.]

Tekoälyä on kahta päätyyppiä: kapea tekoäly (Narrow AI) ja yleinen tekoäly (General AI). Nämä tekoälyt oppivat ja / tai niitä opetetaan suorittamaan tiettyjä tehtäviä ilman ohjeita. Täydellinen esimerkki on puheen ja kielen tunnistusominaisuus iPhonen virtuaalisessa avustajassa, Sirissä. Toinen esimerkki on itseajavien autojen nääntunnistusjärjestelmä. Kolmas esimerkki on tekoälyt, jotka näyttävät mainoksia katsojille heidän hakuhistoriansa ja Internet-toimintojensa perusteella. [18.]

Kapea tekoäly voi oppia tai sitä voidaan opettaa suorittamaan vain tietyn tehtävän. Kapeat tekoälyt voivat tehdä monia asioita, kuten tulkita videosyötteitä valvontadroneista tai arkipäiväisiä tehtäviä, kuten henkilö- ja yritysasiakirjojen järjestämistä. Ne pystyvät vastaamaan asiakkaiden kysymyksiin ja koordinoimaan muiden AI:den kanssa hotellihuoneen varaamista oikeaan hintaan ja oikeaan paikkaan. Niitä on käytetty myös edistyneissä sovelluksissa, kuten mahdollisten syöpäkasvainten havaitsemisessa röntgenkuvassa, hissien kulumisen havaitsemisessa tai sopimattomien sisältöjen ilmoittamisessa verkossa. [18.]

Toisin kuin kapea tekoäly, joka voi oppia tekemään vain yhden asian, yleisellä AI:llä on ihmisten kaltainen mukautuva älykkyys. Tämän joustavuuden ansiosta yleiset tekoälyt voivat luoda laskentataulukoita, leikata hiuksiasi ja ajaa autoa törmäämättä ihmisiin. Tällaisia tekoälyjä näytetään Terminaattorin kaltaisissa elokuvissa. Toistaiseksi sitä ei kuitenkaan ole vielä olemassa, ja tekoälyasiantuntijat keskustelevat edelleen, tuleeko siitä koskaan totta. [18.]

3.2 Koneoppiminen

Koneoppiminen on prosessi, jonka avulla pyritään luomaan ihmisen älykkyyttä emuloivia tietokonealgoritmeja. Se ammentaa ideoita eri tieteenaloista, kuten tekoälystä, todennäköisyyslaskennasta, tilastoista, tietojenkäsittelytieteestä, informaatioteoriasta,

psykologiasta, ohjausteoriasta ja filosofiasta. Koneoppimisen tärkein ominaisuus on itsenäinen oppiminen sitä ympäröivästä ympäristöstä ilman erillistä ohjausta. [19.]

On olemassa neljä erilaista koneoppimistapaa [20.]:

- Ohjattu koneoppiminen: Malli koulutetaan manuaalisesti luokitetulla tietojoukolla, jonka perusteella kone ennustaa tuloksen.
- Ohjaamaton koneoppiminen: Malli koulutetaan luokittelemattomalla tietojoukolla. Kone ennustaa tuloksen ilman erillistä valvontaa.
- Puoliohjattu koneoppiminen: Malli koulutetaan luokitetulla ja luokittelemattomalla tietojoukolla. Puoliohjatun koneoppimisen päätavoite on hyödyntää kaikkea saatavilla olevaa dataa.
- Vahvistusoppiminen: Malli opetetaan palautepohjaisessa prosessissa. Hyvistä toimenpiteistä palkitaan ja huonoista seuraa rangaistus. Dataa ei olla merkattu vaan malli oppii vain ja ainoastaan kokemuksistaan.

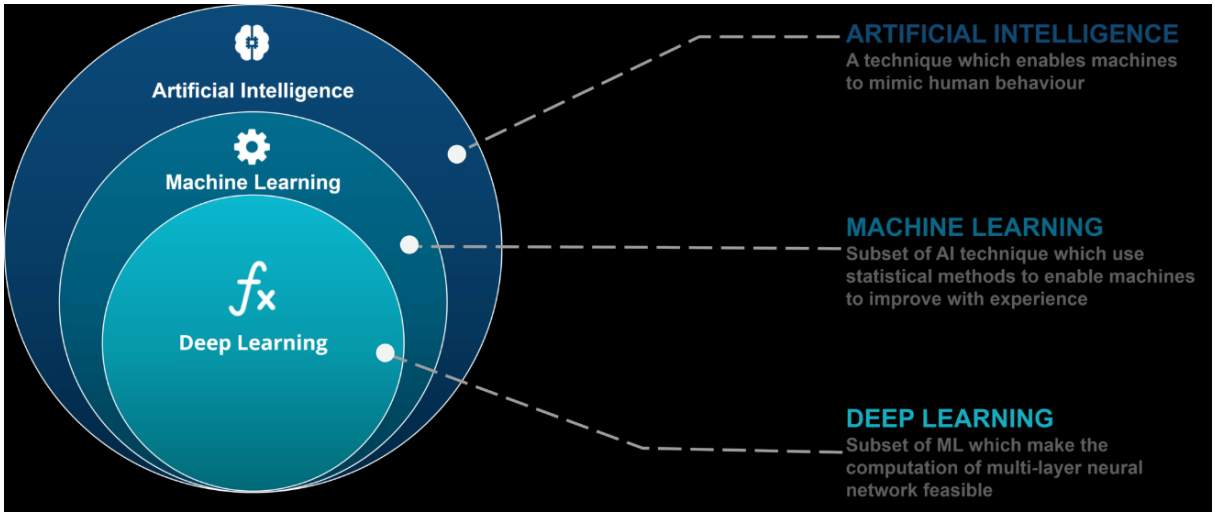
3.3 Syväoppiminen

Syväoppiminen on tekoälyn haara, joka keskittyy massiivisten hermoverkkomallien rakentamiseen, joiden avulla voidaan tehdä tarkkoja johtopäätöksiä datasta. Syväoppiminen on erittäin tehokasta tilanteissa, joissa data on monimutkaista ja saatavilla on paljon tietojoukkoja. [21.]

Nykyään monet huippuluokan kuluttajateknologiayritykset käyttävät syväoppimista. Muun muassa Facebook käyttää syväoppimista analysoidakseen tekstiä verkkokeskusteluissa. Google, Baidu ja Microsoft käyttävät syväoppimista kuvahaussa ja konekääntämisessä. Kaikissa moderneissa älypuhelimissa on syväoppimisjärjestelmät, joita käytetään muun muassa puheentunnistukseen ja myös kasvojen poistoon digikameroissa. [21.]

Terveydenhuollon alalla syväoppimista käytetään lääketieteellisten kuvien, kuten röntgen-, CT- ja MRI-skannausten käsittelyyn ja terveystilojen diagnosointiin. Itseohjautuvia autoja ei olisi olemassa ilman syväoppimista, jota tarvitaan muun muassa lokalisointiin, kartoittamiseen, ympäristön havaitsemiseen, kuljettajan tilan seurantaan sekä liikkeen suunnitteluun ja ohjaamiseen. [21.]

Datatieteen eri osa-alueita käydään läpi alla olevassa kuvassa 4.



Kuva 4. Datatieteen eri osa-alueet [22].

4 Koneoppiminen sulautetussa ympäristössä

4.1 Käyttötarkoitukset

Valmistus- ja teollisuusympäristöissä sulautetun tekoälyn ja esineiden internetin yhdistäminen voi johtaa laitteiden ennakoivaan ylläpitoon, toiminnan tehostamiseen, tuotteiden ja palvelujen parantumiseen sekä riskienhallintaan. [23.]

Lisäksi sulautettujen tekoäly- ja IoT-laitteiden sovelluksia on runsaasti muissa ympäristöissä, ja niillä on vaikutuksia muun muassa turva- ja valvontajärjestelmiin, älykkäisiin koteihin ja kaupunkeihin, sekä tieteelliseen tutkimukseen ja terveydenhuoltoon. [23.]

4.2 Etu perinteiseen koneoppimiseen verrattuna

Perinteisesti monimutkaisia tekoälylaskelmia, kuten hakukonetulosten tuottamista, suoritettiin pilvessä olevassa palvelinkeskuksessa. Tekoälymallien toteuttaminen grafiikka-prosessointiyksiköissä ja järjestelmäpiireissä on vähemmän riippuvainen pilvestä tekoälyn tietojenkäsittelyssä. Sulautetun tekoälyn avulla laitteet voivat ajaa tekoälymalleja laitetasolla ja sitten käyttää tuloksia suoraan sopivan tehtävän tai toiminnon suorittamiseen. Pilvi on edelleen hyödyllinen tiedon tallennuksen näkökulmasta, sillä tiedot voidaan tallentaa tilapäisesti laitetasolle ja lähettää lopulta pilvipalvelimelle säilytettäväksi. Edut ovat muun muassa alhaisemmat tiedonsiirtokustannukset, tietoturva sekä luotettavuus. [23.]

Sulautettu tekoäly **alentaa tiedonsiirtokustannuksia**. Tekoälyalgoritmit vaativat suuren määrän dataa mallien analysointiin ja kouluttamiseen. Tämän vuoksi tarvitaan suuri kaistanleveys, jotta data saadaan siirrettyä pilveen tai muihin palvelinkeskuksiin. Sulautetun tekoälyn avulla laitteista tulee itsenäisiä, mikä johtaa siihen, että laite tarvitsee vähän tai ei ollenkaan kaistaa. [24.]

Sulautettu tekoäly tarjoaa paremman **tietoturvan**. Anturit ja tallennuslaitteet tuottavat arkaluontoista dataa ja tämän vuoksi ne aiheuttavat tietosuojaongelmia. Näiden arkaluontoisten tietojen jakaminen Internetin välityksellä lisää tietosuojaloukkauksen riskiä. Tietojen käsittely sulautetun laitteen sisällä ja tiedonsiirron välttäminen pienentää merkittävästi tietosuojaloukkauksen todennäköisyyttä. [24.]

Sulautettu tekoäly tarjoaa paremman **luotettavuuden**. Laite, joka pystyy käsittelemään tietoja paikallisesti, on vähemmän altis vioille. Tämä on äärimmäinen tärkeää monille alan erikoistyökaluille tai herkille laitteille, joista käyttäjät ovat riippuvaisia. [24.]

5 Sulautettu tekoäly (Embedded Artificial Intelligence)

Sulautettu tekoäly viittaa tekoälyn ja konekielen tekniikoiden käyttöön sulautetuissa järjestelmissä paikallisesti kaapattujen tietojen analysoinnin mahdollistamiseksi. Yhdistettynä laitteen sisäiseen reunalaskentaan sekä pilvessä tapahtuvaan sumulaskentaan sulautettu tekoäly vähentää merkittävästi viivettä, mahdollistaa syvemmän ennakoivan analytiikan ja lyhentää reagointiaikaa vakavissa vikatapauksissa. [25.]

Koska datan tuottaminen ja sen hankita on alati kasvava ala, on tullut yhä tärkeämmäksi, että älykkäät reunalaitteet käyttävät tekoälymalleja, jotka pystyvät mukautuvaan ja asteittaiseen oppimiseen tietyissä tehtävissä. [25.]

Tietojen määrän kasvaessa tekoälymallin kehittäminen vaatii yhä enemmän laskentakapasiteettia. Tyypillisissä sulautetuissa tekoälysovelluksissa laskenta- ja muisti-intensiiviset osat, kuten mallin koulutus, siirretään usein pilvessä oleviin tehokkaisiin grafiikkasuorittimiin tai laskentayksiköihin. Vain tietyt kevyet päättelymallit otetaan käyttöön reunalaitteissa, kuten IoT-laitteissa tai mobiililaitteissa. [25.]

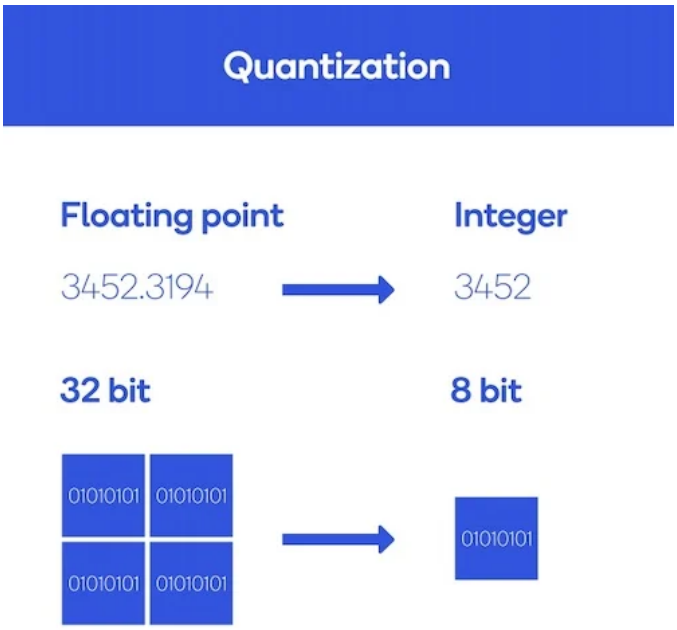
6 Mallin optimoiminen sulautetuissa tekoälysovelluksissa

Monet käytännön sovellukset vaativat laitteessa tapahtuvia reaaliaikaisia käsittelyominaisuuksia. Esimerkiksi kodin turvakameroiden tekoälyn on kyettävä käsittelemään ja varoittamaan omistajaa, mikäli tuntematon henkilö yrittää päästää sisään kiinteistöön. Suurin haaste nykyajan huippuluokan tekoälyn käytössä ovat laitteiden resurssirajoitukset, kuten rajallinen muisti ja prosessointiteho. Hyvin toimivien syväoppimismallien koko on suuri. Tämä aiheuttaa ongelmia, kun kyseistä syväoppimismallia yritetään ottaa käyttöön laitteissa, joilla on rajalliset resurssit. Mitä suurempi malli, sitä enemmän tallennustilaa se tarvitsee. Lisäksi suurempi malli vaatii korkeamman päättelyjakson ja kuluttaa enemmän energiaa päättelyn aikana. Vaikka malli toimisi hyvin hallitussa laboratorioympäristössä, sitä ei välttämättä voida käyttää tosielämän sovelluksissa ja laitteissa. Tämän vuoksi jää vain yksi vaihtoehto: mallin koon kutistaminen. Tämä puolestaan aiheuttaa uuden ongelman: mallin pitää toimia reunalaitteen speksien rajoissa ilman, että mallin tarkkuus kärsii. Näin olleen ei riitä, että malli kykenee toimimaan laitteessa. Mallin tarkkuuden ja nopeuden on pysyttävä samana. [26.]

Seuraavissa kappaleissa käsitellään kolme yleisintä tapaa mallien optimoimiseksi.

6.1 Kvantisointi

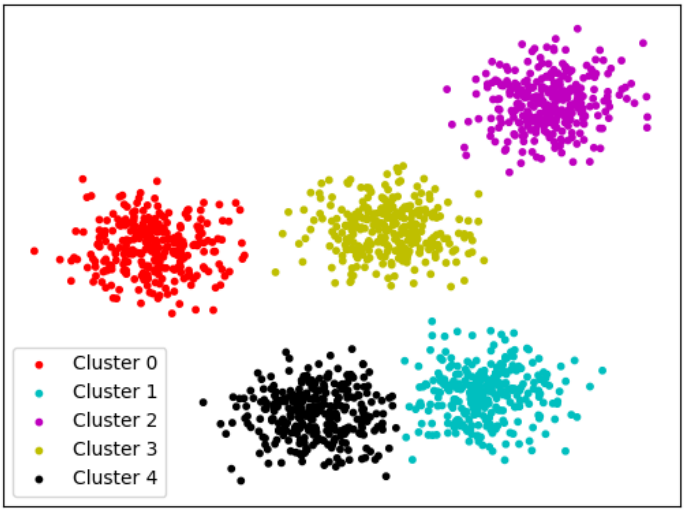
Kvantisointi on prosessi, jolla vähennetään painojen, poikkeamien ja aktivointien tarkkuutta. Tämä johtaa siihen, että ne kuluttavat vähemmän muistia. Esimerkiksi neuroverkko, jonka parametrit ovat 32-bittisiä liukulukuja, muunnetaan 8-bittisiksi kokonaisluvuksi. Tämä pienentää mallin kokoa neljäsosaan alkuperäisestä. [27.] Kuten alla olevasta kuvasta 5 näkyy, tarkkuuden menetys on minimaalinen.



Kuva 5. Kvantisointi käytännössä [27].

6.2 Klusterointi

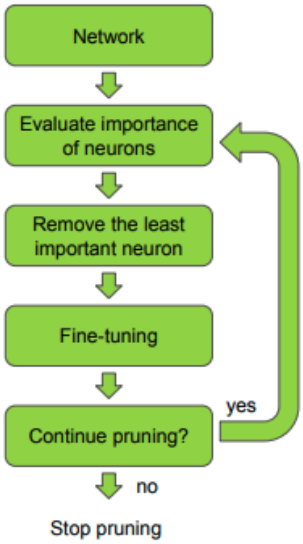
Klusterointi toimii ryhmittelemällä mallin jokaisen kerroksen painot ennalta määritettyyn määrään klustereita. Tämä vähentää yksilöllisten painoarvojen määrää mallissa, mikä vähentää sen monimutkaisuutta. Tämän seurauksena klusteroituja malleja voidaan pakata tehokkaammin, mikä tarjoaa karsimisen kaltaisia käyttöönottoetuja. [28.] Alla olevassa kuvassa 6 on esimerkki klusteroinnista.



Kuva 6. Klusterointi käytännössä [29].

6.3 Karsiminen

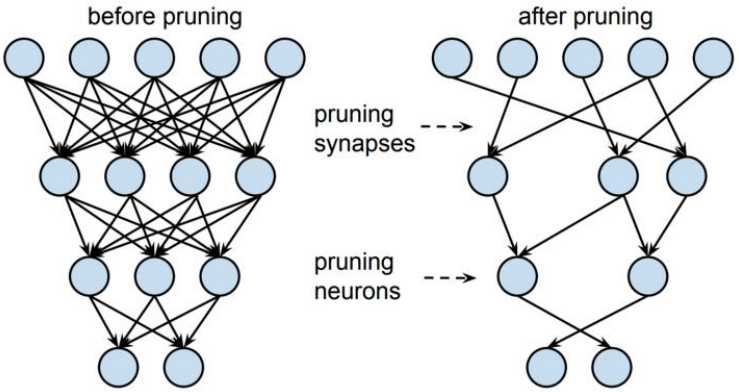
Karsiminen on prosessi, jossa poistetaan painoyhteyksiä verkosta päättelyn nopeuden lisäämiseksi ja mallin tallennuskoon pienentämiseksi. Yleensä neuroverkot ovat erittäin yliparametrisoituja. Verkon karsiminen voidaan ajatella käyttämättömien parametrien poistamisena yliparametrisoidusta verkosta. [30.] Tämä havainnollistetaan alla olevassa kuvassa 7.



Kuva 7. Karsiminen vaiheittain [31].

Synapsien ja neuronien karsiminen

Vähemmän tärkeät synapsit ja neuronit karsitaan (nollataan). Tämä nopeuttaa mallin toimintaa ja vie vähemmän muistia ja suorituskykyä. Mallin tarkkuus kuitenkin kärsii. Tämä havainnollistetaan alla olevassa kuvassa 8.



Kuva 8. Synapsien ja neuronien karsiminen [32] .

7 TinyML

7.1 TinyML:n perusteet

TinyML helpottaa koneoppimisen suorittamista sulautetuissa reunalaitteissa, joissa on hyvin vähän prosessointitehoa ja muistia. [33.] Tällaisten koneoppimista käyttävien järjestelmien virrankulutuksen tulisi olla alle muutaman milliwatin. Tyypillisesti TinyML mahdollistaa IoT-pohjaisten sulautettujen reunalaitteiden siirtymisen alhaisemman tehon järjestelmiin yhdistämällä kehittyneitä virranhallintamoduuleja. Tällaisen järjestelmän tulisi hyödyntää laitteistokiihdytystä. [34.]

Lisäksi koneoppimisen suorittamisessa TinyML-skenaariossa olevan ohjelman tulisi olla mahdollisimman kompakti virransäästösyistä. TinyML-järjestelmien tulisi erikoistua erilaisten koneoppimismallien optimointiin, koska ne tarjoavat paremman tarkkuuden resursseja säästävien rajoitusten vuoksi. TinyML-järjestelmän on täytettävä seuraavat vaatimukset [34.]:

- Sen on kyettävä suorittamaan reunalaitteiden oppimismalleja.
- Se mahdollistaa akkukäyttöisten sulautettujen reunalaitteiden käytön.
- Sen voi skaalata biljooniin anturikäyttöisiin halvoihin sulautettuihin laitteisiin.
- Koodi on muutaman kilotavun kokoinen ja se voidaan tallentaa laitteen RAM-muistiin.

Nykypäivän koneoppimislaitteita käytetään julkisissa pilvissä sekä yksityisissä tiloissa. Organisaatiot käyttävät valmiita käyttöön otettuja malleja erilaisissa teollisissa sovelluksissa. [34.]

Riippuvuus pilvipohjaisista koneoppimispalveluista aiheuttaa muutamia haasteita, kuten [34.]:

- Valtavan energiankulutuksen.
- Yksityisyysongelmia.
- Verkon ja prosessoinnin latenssia.
- Monenlaisia luotettavuusongelmia.

7.2 TinyML:n rajoitukset

Tällä hetkellä suunnitellun TinyML:n kasvua rajoittaa neljä keskeistä tekijää [35.]:

- Energia: Olemassa olevat IoT-pohjaiset sulautetut reunalaitteet vaativat vähintään 10–100 mAh akun itsenäiseen työskentelyyn. Näin ollen reunalaitteiden virtalähteiden on kyettävä tuottamaan tarpeeksi energiaa erilaisiin koneoppimistehtäviin.
- Prosessorikapasiteetti: Suurimmassa osassa pienistä reunalaitteista on 10–1000 MHz:n kellotaajuus. Se voi estää monimutkaisia oppimismalleja toimimasta tehokkaasti laitteen reunalla.
- Muisti: Olemassa olevissa pienissä reunaympäristöissä on keskimäärin alle 1 Mt sisäistä flash-muistia ja 1000 KB SRAM-muistia.
- Kustannukset: Vaikka yksittäisten laitteiden kustannukset ovat alhaiset, mikäli laitteet otetaan käyttöön massiivisessa mittakaavassa, voi se aiheuttaa valtavia kokonaiskustannuksia. Tällaisten ongelmien poistaminen on välttämätöntä, jotta TinyML menestyy edullisissa reuna-alustoissa.

7.3 TinyML:n määritelmä

TinyML voidaan määritellä seuraavasti: "Koneoppimistietoinen arkkitehtuurit, viitekehykset, tekniikat, työkalut ja lähestymistavat, jotka pystyvät suorittamaan analyyttisiä tunnistusmenetelmiä laitteen sisällä mikrowatin tai sitä pienemmällä tehoalueella, samalla kun se kohdistuu pääasiassa akkukäyttöisiin sulautettuihin reunalaitteisiin, jotka soveltuvat toteutettaviksi suuressa mittakaavassa, mieluiten IoT:n tai langattoman anturiverkon alueella". [34.]

TinyML voidaan kuvitella kolmen avainelementin, ohjelmiston, laitteiston ja algoritmien kokoonpanona. TinyML voidaan sijoittaa Linuxiin, sulautettuun Linuxiin ja pilvipohjaisiin ohjelmistoihin, joissa voidaan käyttää alkuperäisiä TinyML-sovelluksia. Laitteisto voi sisältää IoT-laitteita joko laitteistokiihdytyksen kanssa tai ilman sitä. Tällaiset laitteet voivat perustua muistin sisäiseen laskemiseen, analogiseen laskentaan ja neuromorfiseen laskentaan paremman oppimiskokemuksen saavuttamiseksi. TinyML-järjestelmän algoritmien tulee olla uusia, jotta kilobitin kokoisia malleja voidaan ottaa käyttöön resursseja säästävissä reunalaitteissa. Paremmat pakkaus- ja kvantisointimenetelmät ovat välttämättömiä tässä yhteydessä. [34.]

7.4 TinyML reunalaitteen sisällä

Raunalla toimivan kanavan toiminta alkaa antureista, jotka keräävät raakaa dataa ja jotka tarjoavat signaalisuodattimet. Signaalisuodattimet suodattavat tiedot ominaisuuksien ulottuvuuden perusteella. Jos data on esimerkiksi aikasarjasuuntautunut, lasketaan aikasarjan ominaisuudet. Vaihtoehtoisesti myös spektriominaisuudet voidaan laskea. Näytteitä säilytetään sitten FIFO (First In First out)-tietorakenteessa erittäin lyhyen ajan. Jos data on aikasarjamuodossa, niin stationaarisuusluokittajalla tarkistetaan, mikäli data seuraa stationäärisiä attribuutteja. Seuraavassa vaiheessa pyritään tasoittamaan IoT-pohjainen yhteys pitkän aikavälin mallimuistiin, joka puolestaan kommunikoi kuvioluokittajan kanssa joko sääntöpohjaista käsittelyä tai klusteriproseduuria varten sovelluskontekstin mukaan. [36.]

7.5 TinyML-työkalut

TinyML vaatii useita laitteistospesifikaatioita, kirjastoja ja ohjelmistoalustoja ennusteiden hyödyntämiseksi. Esittelen lyhyesti laitteisto- ja ohjelmistotyökaluja, joita mahdollisesti voidaan käyttää TinyML-ympäristössä.

Alla olevassa taulukossa 1 olevat elektroniikka-alustat ovat TinyML-yhteensopivia:

Apollo3	STM32F Discovery
ST IoT Discovery	ECM3532 AI Sensor Neuro sensor processor
Arduino Nano 33 BLE Sense	OpenMV Cam H7 Plus
Himax EW-I Plus	Thunderboard Sense 2
Sony Spresense TinyML Board	Arduino Portenta H7
Raspberry Pi 4B	Nvidia Jetson Nano
Laupadn2352P13	ESP-EYE
Laupadn2352P1	AI-deck 1
GAP8	Agora Product Development Kit
GAP9	MKR Video 4000
Seed Wio Terminal	Nordic Semi nDK284 nRF5

Pico4ML BLE Nicla Sense ME	Nordic Semi Thingy 91 XCore.ai

Taulukko 1: TinyML-yhteensopivat elektroniikka-alustat [34].

Nykyisillä markkinoilla on saatavilla monia muita alustoja, joiden soveltuvuutta TinyML:n kanssa voidaan tutkia. Seuraavassa taulukossa vertaillaan edellä mainittujen laitteistoalustojen prosessorin, suorittimen kellotaajuuden, flash-muistin, SRAM-koon, tehon tai jännitteen kulutuksen, liitettävyyden, antureiden tai liittimien ja tuotekehittäjän suhteen. Huomataan, että useimmassa laitteistokortissa on alle 100 MHz:n prosessoritaajuus, keskimäärin alle 1 Mt:n flash-muistia ja alle 1 Mt:n SRAM-muistia. Bluetooth (BLE) ja Wi-Fi ovat enimmäkseen valittuja yhteystekniikoita. Havaitaan, että useimmissa alustoissa voi olla useita sisäisiä antureita, kuten kiihtyvyys-, lämpötila-, kosteus-, mikrofoni-, gyroskooppi-, ilmanpaine-, eletunnistin-, valoanturi, ilmanlaatu- ja kameramittari. Tällaisten alustojen virrankulutus on pääosin mW:n alueella. Useimpia laitteita voidaan käyttää Li-Po- ja kolikkoakuilla tavallisen tasavirtalähteen lisäksi. [34.]

8 Koneoppimisen toteutus sulautetussa laitteessa – esimerkkisovellus

Tämän esimerkkisovelluksen tarkoituksena on harjoitella tietojen keräämistä, neuroverkon koulutusta Edge Impulse -sivustoa hyväksi käyttäen sekä reaaliaikaista testaamista. Tämän lisäksi luodaan ääniohjattu kaukosäädin käyttämällä Syntiant TinyML -korttia. Kaukosäädin vastaa kahteen komentoon: kun käyttäjä sanoo sanan ”Go”, syttyy vihreä valo, ja kun käyttäjä sanoo sanan ”Stop”, syttyy punainen valo. Tässä esimerkkisovelluksessa on kolme vaihetta: koneoppiminen, käyttöönotto sekä reaaliaikainen testaus.

8.1 Koneoppiminen

Tässä osiossa opitaan luomaan hermoverkkomalli, joka toimii Syntiant TinyML -levyllä. Ensin kloonataan Edge Impulse – sivuston tarjoama valmis projekti. Tämän jälkeen tarkastellaan kopioitu projekti ja varmistutaan siitä, että kaikki tarvittavat osiot on kloonattu onnistuneesti. Kun data on saatu verifioitua, luodaan esimerkki ”Go”- ja ”Stop”-luokittimista. Tämän jälkeen aloitetaan impulssisuunnittelu. [37.]

Ensin luodaan impulssi, jonka jälkeen poistetaan oletuksena olevat Syntiatin äänilohko ja Keraksen hermoverkkolohko. Tämän jälkeen lisätään uusi Keraksen käsittelylohko ja Syntiatin oppimislohko. Kun edellä mainitut lohkot on määritelty oikein, luodaan ominaisuudet, joiden avulla ohjelma generoi kolmiulotteisen tietokannan, joka näyttää klusterit ja niiden erottelun luokittelijoilla. Tämän jälkeen valitaan NN (nearest neighbours)-luokitus ja aloitetaan mallin harjoittaminen. [37.]

8.2 Käyttöönotto

Tässä osiossa opitaan, miten hermoverkkomalli otetaan käyttöön Syntiant TinyML -levylle. Aluksi malli otetaan käyttöön, minkä jälkeen etsitään Syntiant TinyML -levyn parametrit. Tämän jälkeen tietokoneelle asennetaan Arduino CLI (command-line interface), jonka avulla neuroverkko saadaan siirrettyä Syntiant TinyML -levylle. Kun tämä on tehty, Syntiant TinyML -levy yhdistetään tietokoneeseen ja neuroverkko asennetaan levylle. [37.]

8.3 Reaaliaikainen testaus

Tässä osiossa neuroverkkoa, joka on asennettu Syntiant TinyML -levylle, testataan reaaliajassa. Mikäli edellä mainitut vaiheet on saatu suoritettua onnistuneesti, Syntiant TinyML -kaukosäädin on valmis. Nyt sitä voidaan testata reaaliaikaisesti: kun käyttäjä sanoo sanan ”Go”, syttyy vihreä valo, ja kun käyttäjä sanoo sanan ”Stop”, syttyy punainen valo. Jos käyttäjä ei sano mitään tai sanoo sanan, joka ei ole ”Go” tai ”Stop”, mitään ei tapahdu. [37.]

9 Yhteenveto

Työn tavoitteena oli selvittää, miten koneoppimista voidaan soveltaa sulautetussa ympäristössä, tutkia tekoälyn historiaa, käydään läpi tekoälyn eri osa-alueita, syventyä TinyML:ään, sekä tehdä esimerkkiharjoitus, joka liittyy sulautettuun tekoälyyn. Kaikki edellä mainitut tavoitteet saatiin toteutettua onnistuneesti.

Työn edistyminen oli hidasta mutta tasaista. Teorian tutkimukseen ja tiedonhankintaan meni paljon aikaa. Paremmalla ajankäytöllä työ olisi voitu saada valmiiksi aikaisemmin.

Opinnäytetyötä voidaan jatkokehittää muun muassa tutkimalla muita sulautetun tekoälyn osa-alueita, tekemällä vaativampia esimerkkiharjoituksia ja tekemällä yksinkertaisia omia sovelluksia.

Lähteet

1. McCulloch, W.S. Pitts, W. A logical calculus of the ideas immanent in nervous activity. [Internet]. The Bulletin of Mathematical Biophysics, 1943. Saatavilla: https://doi.org/10.1007/BF02478259

2. Hebb, D. O. The Organization of Behavior: A neuropsychological theory. John Wiley & Sons Inc 1949.

3. Turing, A. M. Computing Machinery and Intelligence. Oxford University Press 1950.

4. Newell, A. Shaw, J.C. Herbert S. Logic Theorist. [Internet] 1955 [viitattu 23.5.2023]. Saatavilla: https://ahistoryofai.com/logic-theorist

5. McCarty, J. Minsky, M. L. Rochester, N. Shannon, C. E. A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence. [Internet]. 1956 [viitattu 23.5.2023]. Saatavilla: https://home.dartmouth.edu/about/artificial-intelligence-ai-coined-dartmouth

6. Weizenbaum, J. Computer Power and Human Reason: From Judgment to Calculation. W. H. Freeman and Company 1976.

7. Kato, I. Ohteru, S. Kobayashi, H. Shirai, K. Uchiyama, A. On Theory and Practice of Robots and Manipulators. [Internet]. Springer Berlin, Haidelberg, 1974. Saatavilla: https://doi.org/10.1007/978-3-662-40393-8

8. Kato, I. WABOT -WAseda roBOT-. [Internet]. 1985 [viitattu 23.5.2023]. Saatavilla: https://www.humanoid.waseda.ac.jp/booklet/kato_2.html

9. Feigenbaum, A. E. Expert Systems: Principles and Practice. [Internet]. 1980 [viitattu 23.5.2023]. Saatavilla: https://www.veloxitiai.com/documents/Feigenbaum-EXPERT-SYSTEMS-PRINCIPLES-AND-PRACTICE.pdf

10. First National Conference on Artificial Intelligence. [Internet]. 1980 [viitattu 23.5.2023]. Saatavilla: https://dblp.org/db/conf/aaai/aaai80.html

11. Crevier, D. AI: The Tumultuous History of the Search for Artificial Intelligence. BasicBooks 1993.

12. Pandolfini, B. Kasparov and Deep Blue: The Historic Chess Match Between Man and Machine. Touchstone 1997.

13. iRobot Introduces Roomba™ Intelligent FloorVac - The First Automatic Floor Cleaner In The U.S. [Internet]. 2002 [viitattu 23.5.2023]. Saatavilla: https://media.irobot.com/2002-09-18-iRobot-Introduces-Roomba-Intelligent-FloorVac-The-First-Automatic-Floor-Cleaner-In-The-U-S

14. A Computer Called Watson. [Internet]. 2011 [viitattu 23.5.2023]. Saatavilla: https://www.ibm.com/ibm/history/ibm100/us/en/icons/watson/

15. Rougeau, M. Google IO 2012: Google introduces Siri-killer Google Now. [Internet]. 2012 [viitattu23.5.2023]. Saatavilla: https://www.techradar.com/news/software/operating-systems/google-io-2012-google-introduces-siri-killer-google-now-1087130

16. Aamoth, D. Interview with Eugene Goostman, the Fake Kid Who Passed the Turing Test. [Internet]. 2014 [viitattu 23.5.2023]. Saatavilla: https://time.com/2847900/eugene-goostman-turing-test/

17. De Vynck, G. IBM's Debating AI Is Here to Convince You That You're Wrong. [Internet]. 2018 [viitattu 23.5.2023]. Saatavilla: https://www.bloomberg.com/news/articles/2018-06-19/ibm-s-debating-ai-is-here-to-convince-you-that-you-re-wrong?leadSource=uverify%20wall

18. Ramar, S. Artificial Intelligence: How It Changes the Future. Amazon Digital Services LLC; 2019.

19. El Naqa, I. Murphy, M. What Is Machine Learning? [Internet]. Springer International Publishing Switzerland, 2015. Saatavilla: http://doi.org/10.1007/978-3-319-18305-3_1

20. Types of Machine Learning. [Internet]. [viitattu 23.5.2023]. Saatavilla: https://www.javatpoint.com/types-of-machine-learning

21. Kelleher, J. Deep Learning. The MIT Press 2019.

22. Castello, A. Understanding AI vs Machine Learning vs Deep Learning. [Internet]. 2019 [viitattu 23.5.2023]. Saatavilla: https://intelligentproduct.solutions/blog/ai-machine-learning-deep-learning/

23. Atwell, C. Fundamentals: What is embedded AI? [Internet]. 2021 [viitattu 23.5.2023]. Saatavilla: https://www.fierceelectronics.com/electronics/what-embedded-ai

24. Chawla, Y. Embedded Artificial Intelligence for Business Purposes. [Internet]. 2022 [viitattu 23.5.2023]. Saatavilla: https://dac.digital/embedded-artificial-intelligence-for-business-purposes/

25. Mantri, V. Embedded AI. Algorithm, Model, and Hardware. [Internet]. 2021 [viitattu 23.5.2023]. Saatavilla: https://insights.ltts.com/story/embedded-ai-algorithm-model-and-hardware/page/1

26. Pokhrel, S. 4 Popular Model Compression Techniques Explained. [Internet]. 2022 [viitattu 23.5.2023]. Saatavilla: https://xailient.com/blog/4-popular-model-compression-techniques-explained/

27. Hertz, J. Neural Network Quantization: What Is It and How Does It Relate to TinyML? [Internet]. 2022 [viitattu 23.5.2023]. Saatavilla: https://www.allaboutcircuits.com/technical-articles/neural-network-quantization-what-is-it-and-how-does-it-relate-to-tiny-machine-learning/

28. TensorFlow Model optimization. [Internet]. 2021 [viitattu 23.5.2023] Saatavilla: https://www.tensorflow.org/lite/performance/model_optimization#types_of_optimization

29. Shtar, G. Margel, S. Clustering and Dimensionality Reduction: Understanding the "Magic" Behind Machine Learning. [Internet]. 2017 [viitattu 23.5.2023]. Saatavilla: https://www.imperva.com/blog/clustering-and-dimensionality-reduction-understanding-the-magic-behind-machine-learning/

30. ODSC Community. What is Pruning in Machine Learning? [Internet]. 2022 [viitattu 23.5.2023]. Saatavilla: https://opendatascience.com/what-is-pruning-in-machine-learning/

31. Molchanov, P. Tyree, S. Karras, T. Aila, T. Kautz, J. Pruning Convolutional Neural Networks for Resource Efficient Inference. [Internet]. 2017 [viitattu 23.5.2023]. Saatavilla: https://arxiv.org/pdf/1611.06440.pdf

32. Souvik, P. Pruning in Deep Learning Model. [Internet]. 2020 [viitattu 23.5.2023]. Saatavilla: https://medium.com/@souvik.paul01/pruning-in-deep-learning-models-1067a19acd89Fscein

33. Jonakiram, MSV. How TinyML Makes Artificial Intelligence Ubiquitous. [Internet]. 2020 [viitattu 23.5.2023]. Saatavilla: https://www.forbes.com/sites/janakirammsv/2020/11/03/how-tinyml-makes-artificial-intelligence-ubiquitous/?sh=45128c6a7622

34. Ray, P. P. A review on TinyML: State-of-the-art and prospects. [Internet]. Journal of King Saud University – Computer and Information Sciences, 2022. Saatavilla: https://doi.org/10.1016/j.jksuci.2021.11.019

35. Johnny, F. Knutsson, F. TinyML Talks: CMSIS-NN & Optimizations for Edge AI. [Internet]. 2021 [viitattu 23.5.2023]. Saatavilla: https://cms.tinyml.org/wpcontent/uploads/emea2021/tinyML_Talks_Felix_Johnny_Thomasmathibalan_and_Fredrik_Knutsson_210208.pdf

36. Eroma, A. TinyML Talks: ”Unsupervised collaborative learning technology at the Edge for industrial machine vendors”. [Internet]. 2020 [viitattu 23.5.2023]. Saatavilla: https://cms.tinyml.org/wp-content/uploads/talks2020/tinyML_Talks_Alexander_Eroma_200428.pdf

37. Syntiant TinyML Tutorial for Windows. [Internet]. 2021 [viitattu 23.5.2023]. Saatavilla: https://www.syntiant.com/_files/ugd/64a391_8acb0f3c9faa49c9a9257f1a01beb95a.pdf