

Ville Salmi

# FPGA-KIIHDYTTIMIEN KÄYTTÖ KONE- OPPIMISESSA

Kandidaatintyö  
Informaatioteknologian ja viestinnän tiedekunta  
Tarkastaja: Sakari Lahti  
Huhtikuu 2023

# TIIVISTELMÄ

Ville Salmi: FPGA-kiihdyttimien käyttö koneoppimisessa  
Kandidaatintutkielma  
Tampereen yliopisto  
Tieto- ja sähkötekniikan kandidaattiohjelma  
Huhtikuu 2023

---

Koneoppimisen hyödyntäminen on nykyään suosittua monessa eri sovelluskohteessa. Erityisen suosittuja ovat neuroverkkomallit, joiden tarkkuus ja koko kasvavat vuosi vuodelta. Entistä suuremmat koneoppimismallit luovat tarpeen kasvattaa malleja käsittelevien laitteistojen laskentakapasiteettia, mikä yleensä toteutetaan käyttämällä laitteistokiihdyttämiä. Tässä tutkielmassa perehdytään kirjallisuuslähteiden avulla FPGA-pohjaisten koneoppimiskiihdyttimien rakenteeseen ja ominaisuuksiin, joita työssä vertaillaan muihin yleisesti käytettyihin kiihdytinteknologioihin. Työssä myös esitellään lyhyesti tutkielman aiheelle keskeisiä osa-alueita, kuten FPGA-piirit, koneoppiminen ja laitteistokiihdytys.

FPGA-pohjaiset koneoppimiskiihdyttimet koostuvat usein itse FPGA-piiriin lisäksi kiihdyttimen toimintaa ohjaavasta keskusyksiköstä, keskusyksikön ja FPGA:n välisestä kommunikointiväylästä sekä ulkoisista muistikomponenteista. FPGA-piirille toteutettu kiihdytinsykli koostuu usein monesta erillisestä laskentayksiköstä, jotka suorittavat laskentaa rinnakkain. Tällaista rakennetta hyödyntää esimerkiksi neuroverkkojen kiihdyttämiseen suunniteltu DLAU-kiihdytin.

FPGA-pohjaisilla kiihdyttimillä saavutetaan suuri rinnakkaisuus, jonka avulla voidaan kiihdyttää neuroverkkorakenteiden laskentaa tehokkaasti. FPGA-piirit tarjoavat myös uudelleenohjelmitavuudellaan paljon joustavuutta kiihdyttimen suunnitteluun, sillä FPGA-kiihdytin voidaan helposti muokata laskemaan esimerkiksi eri kokoisia neuroverkkoja. FPGA-piirit kuluttavat myös vähän virtaa, minkä vuoksi niitä voidaan hyödyntää sellaisissa kiihdytinsovelluksissa, joissa energiankulutus on pientä. FPGA-koneoppimiskiihdyttimissä haasteellista on kuitenkin suurien koneoppimismallien käsittely, sillä FPGA-piirien laskenta- ja muistiresurssit ovat hyvin rajallisia. Miljoonia parametrejä sisältävien mallien tallentamiseksi FPGA-pohjaisissa koneoppimiskiihdyttimissä joudutaan käyttämään piiriin ulkopuolisia muistikomponentteja, mikä heikentää suorituskykyä. Haasteita kiihdyttimien suunnittelussa aiheuttaa myös yleiskäyttöisiä ohjelmointikieliä alemmaa abstraktiotasoa olevien laitteistonkuvauskielien käyttö.

ASIC-koneoppimiskiihdyttimiin verrattuna FPGA-kiihdyttimet ovat suoritusteholtaan ja energiatehokkuudeltaan huonompia. ASIC-piirit eivät kuitenkaan ole FPGA-piirien tavoin uudelleenohjelmitavia, mikä heikentää ASIC-kiihdyttimien joustavuutta. Grafiikkasuorittimilla toteutetut koneoppimiskiihdyttimet ovat FPGA-kiihdyttimiin verrattuna paljon tehokkaampia monimutkaisten koneoppimismallien laskennassa. Grafiikkasuoritinpohjaiset koneoppimiskiihdyttimet kuluttavat kuitenkin hyvin paljon virtaa, minkä vuoksi ne eivät sovellu FPGA-kiihdyttimien tavoin vähäisen energiankulutuksen sovelluksiin.

Tulevaisuudessa FPGA-koneoppimiskiihdyttimien suosion odotetaan kasvavan. Vähäistä energiankulutusta vaativissa tulevaisuuden sovelluksissa, kuten sulautetuissa järjestelmissä ja pilvipalveluissa, käytettävien koneoppimismallien laskenta voidaan suorittaa joustavasti FPGA-pohjaisten koneoppimiskiihdyttimien avulla. FPGA-koneoppimiskiihdyttimet tulevat olemaan tulevaisuudessa suuressa roolissa myös reaaliaikaista prosessointia vaativissa sovelluksissa, kuten itseajavissa ajoneuvoissa.

Avainsanat: FPGA, koneoppiminen, laitteistokiihdytys, neuroverkot

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

# SISÄLLYSLUETTELO

1. JOHDANTO .....	1
2. FPGA-PIIRIT.....	3
2.1    FPGA-piirin rakenne.....	3
2.2    FPGA-piirin ohjelmitavuus.....	4
2.3    FPGA-piirien käyttökohteet .....	4
3. KONEOPPIMINEN.....	6
3.1    Koneoppimisalgoritmit.....	6
3.2    Neuroverkot ja niiden rakenne .....	7
3.3    Koneoppimisen sovelluskohteet.....	8
4. LAITTEISTOKIIHDYTYS.....	10
4.1    Laitteistokiihdyttimien toimintaperiaate .....	10
4.2    Laitteistokiihdytyksessä käytetyt teknologiat .....	11
4.3    Laitteistokiihdytyksen sovellukset.....	12
5. FPGA-KIIHDYTTIMET KONEOPPIMISESSA .....	13
5.1    FPGA-pohjaisten koneoppimiskiihdyttimien rakenne.....	13
5.2    FPGA-pohjaisten koneoppimiskiihdyttimien hyödyt ja ongelmat.....	16
5.3    FPGA-pohjaisten koneoppimiskiihdyttimien vertailu muihin kiihdytinteknologioihin .....	17
5.4    FPGA-pohjaisten koneoppimiskiihdyttimien merkitys tulevaisuudessa	19
6. YHTEENVETO.....	22
LÄHTEET .....	24

# LYHENTEET JA MERKINNÄT

AES	engl. Advanced encryption standard, salausalgoritmi
AFAU	engl. Activation Function Acceleration Unit, neuroverkon aktivaatiofunktion laskentaa suorittava laskentayksikkö
ASIC	engl. Application-Specific Integrated Circuit, sovelluskohtainen mikropiiri
BRAM	engl. Block Random Access Memory, FPGA-piirille toteutettu hajasaantimuistilohko
CLB	engl. configurable logic block, FPGA-piirin ohjelmoitava logiikka-lohko
CNN	engl. convolution neural network, konvoluutioneuroverkko
CPU	engl. central processing unit, keskusyksikkö
DDR	engl. Double Data Rate, hajasaantimuistityyppi
DLAU	engl. Deep Learning Accelerator Unit, FPGA-pohjainen neuroverkkokiihdytin
DMA	engl. Direct memory access, suora muistihaku
DSP	engl. Digital Signal Processing, digitaalinen signaalinkäsittely
FPGA	engl. Field Programmable Gate Array, uudelleenohjelmoitava porttimatriisi
GOPS	engl. Giga Operations Per Second, suoritustehon mittaamiseen käytetty yksikkö
HDL	engl. hardware description language, laitteistonkuvauskieli
I/O	engl. Input/Output, tulo/lähtö
IoT	engl. Internet of Things, esineiden internet
LUT	engl. lookup table, hakutaulukko
MFU	engl. multifunctional unit, neuroverkkokiihdyttimen yleiskäyttöinen laskentayksikkö
MLP	engl. multi-layer perceptron, moniasteperseptroni
NFU	engl. neural functional unit, neuroverkkokiihdyttimen päälaskentayksikkö
NPU	engl. neural processing unit, Intelin kehittämä FPGA-pohjainen neuroverkkokiihdytin
NRE	engl. Non-recurring engineering, kertaluontoinen tuotekehityskustannus
NVIDIA CUDA	engl. Compute Unified Device Architecture, NVIDIA:n kehittämä ohjelmointialusta grafiikkasuoritinpohjaisille laitteistokiihdyttimille

PCIe	engl. Peripheral Component Interconnect Express, kommunikointiväylätyyppi
PSAU	engl. Part Sum Accumulation Unit, osasummien laskentayksikkö
RNN	engl. Recurrent Neural Network, takaisinkytketty neuroverkko
SRAM	engl. Static Random Access Memory, staattinen hajasaantimuisti
SoC	engl. System On Chip, järjestelmäpiiri
TMMU	engl. Tiled Matrix Multiplication Unit, matriisikertolaskuja suorittava laskentayksikkö
TOPS	engl. Tera Operations Per Second, suoritustehon mittaamiseen käytetty yksikkö
TPU	engl. Tensor Processing Unit, Googlen kehittämä ASIC-pohjainen koneoppimiskiihdytin
UART	engl. Universal Asynchronous Receiver and Transmitter, sarjaliikennepiiri
VHDL	engl. Very High-Speed Integrated Circuit Hardware Description Language, yleisesti käytetty laitteistonkuvauskieli

# 1. JOHDANTO

Teknologia on älykkäämpää kuin koskaan. Erityisesti koneoppimisen kiihtyvä kehitys on tuonut markkinoille entistä älykkäämpiä laitteita ja ohjelmistoja, ja koneoppimisalgoritmit ovat tärkeitä työkaluja massiivisten datamäärien hallinnassa ja sen muokkaamisessa monessa eri sovelluskohteessa [1]. Hyvin merkittävässä roolissa modernissa koneoppimisessa ovat syväoppimisessa käytetyt neuroverkot kuten konvoluutioneuroverkot (CNN, engl. convolution neural network), joiden avulla saadaan toteutettua entistä tarkempia malleja moniin eri koneoppimisen sovelluksiin, kuten esimerkiksi konenäköön ja puheentunnistukseen [2].

Monimutkaisten neuroverkkojen käyttäminen tuo mukanaan kuitenkin haasteita laskennalliselta näkökulmalta, sillä paljon laskentatehoa ja suuren datamäärän käsittelyä vaativien operaatioiden suorittaminen on perinteisellä suorittimella (CPU, engl. central processing unit) hyvin haasteellista. Suorittimen sijaan paljon laskentatehoa vaativissa tehtävissä käytetään usein kyseiseen laskentatarkoitukseen suunniteltuja laitteistokiihdyttimiä, joissa käytetyimpiä teknologioita ovat FPGA-piirit (engl. Field Programmable Gate Array), ASIC-piirit (engl. Application-Specific Integrated Circuit) sekä grafiikkasuorittimet [3].

FPGA-piirit ovat helposti uudelleenohjelmoitavissa ja tarjoavat näin ollen mahdollisuuden muokata FPGA-pohjaisen laitteistokiihdyttimen toiminnallisuutta joustavasti myös jälkikäteen käyttökohteeseen sopivaksi. FPGA-piirit myös käyttävät hyvin vähän virtaa, mikä tekee niistä houkuttelevan teknologiavaihtoehdon sellaisiin sovelluksiin, joissa energiankulutus pyritään pitämään mahdollisimman pienenä. [2] Tässä tutkielmassa tavoitteena on selvittää kirjallisuuslähteitä hyödyntäen, miten juuri FPGA-piireihin pohjautuvia laitteistokiihdyttimiä voidaan hyödyntää koneoppimisessa ja millaisia hyötyjä ja haittoja FPGA-kiihdyttimillä on verrattuna muihin koneoppimisessa käytettyihin kiihdytinteknologioihin.

Tutkielman toisessa luvussa esitellään yleisesti FPGA-piirien perusrakennetta, niiden ohjelmoitavuutta ja käyttökohteita. Kolmannessa luvussa käsitellään koneoppimisen teoreettinen tausta, koneoppimisessa nykyään yleisesti käytettävää neuroverkkomallia ja koneoppimisen eri sovelluskohteita. Neljäs luku esittelee laitteistokiihdytyksen perusidean, siinä käytettävät eri teknologiat sekä muutamia eri laitteistokiihdytyksen sovelluskohteita. Viidennessä luvussa selvitetään, miten FPGA-kiihdyttimiä voidaan hyödyntää

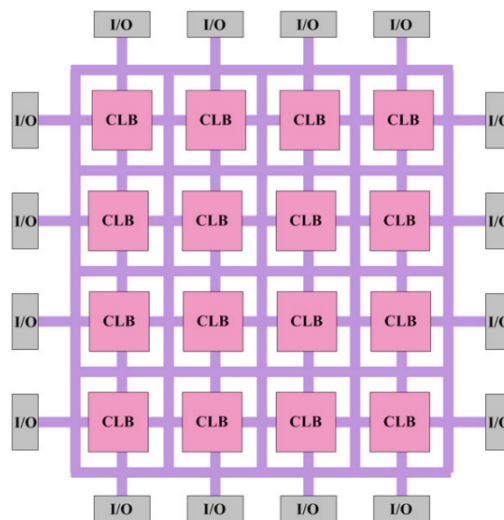
koneoppimisessa, millaisia hyötyjä siitä saavutetaan ja millaisia ongelmia FPGA-pohjaisissa laitteistokiihdyttimissä esiintyy koneoppimissovelluksissa. Viidennessä luvussa myös tutkitaan, miten FPGA-kiihdyttimet vertautuvat muihin koneoppimisessa käytettyihin kiihdytinteknologioihin, ja luvun loppuun myös tehdään katsaus FPGA-koneoppimiskiihdyttimien tulevaisuudennäkymiin ja potentiaalisiin tulevaisuuden sovelluskohteisiin. Tutkielman viimeinen luku tarjoaa yhteenvedon tutkielmassa läpikäytyistä asioista.

## 2. FPGA-PIIRIT

Tämän luvun tarkoituksena on esitellä FPGA-piirien perusrakenne, miten FPGA-piirejä ohjelmoidaan sekä muutamia tärkeitä FPGA-piirien sovellus- ja käyttökohteita.

### 2.1 FPGA-piirin rakenne

FPGA, eli ohjelmoitava porttimatriisi, on digitaalitekniikassa käytetty ohjelmoitava mikropiiri. FPGA-piirien merkittävin ominaispiirre on, että ne ovat helposti uudelleen ohjelmoitavissa, mikä tekee niiden toiminnallisuuden muokkaamisesta jälkikäteen hyvin joustavaa, ja FPGA-piirit tarjoavatkin helposti muokattavissa olevia suunnitteluratkaisuja moniin eri sovelluskohteisiin [4]. FPGA-piirin yksinkertaistettu perusrakenne on esitetty kuvassa 1.



**Kuva 1.** FPGA-piirin perusrakenne [5].

FPGA-piirien toiminnan ydin ovat matriisirakenteen muodostavat logiikkalohkot. Yksinkertaisen logiikkalohkon toiminnallisuuden määrittelee LUT (engl. lookup table). LUT ohjelmoidaan toteuttamaan haluttu looginen operaatio muokkaamalla LUT:ssa olevien SRAM-muistialkoiden (engl. Static Random Access Memory) bittejä. Modernit FPGA-piirit yleensä koostuvat kuvan 1 mukaisista CLB-lohkoista (engl. Configurable Logic Block), jotka yhden logiikkalohkon sijaan sisältävät usean yksinkertaisen logiikkalohkon. FPGA-piirin laajemman toiminnallisuuden määrittelee se, miten matriisirakenteen CLB-lohkot yhdistetään toisiinsa ja I/O-lohkoihin (engl. Input/Output), joita käytetään FPGA-

piirin ja ulkomaailman väliseen kommunikointiin. Lohkojen välinen reititys tehdään ohjelmoimalla, mikä onnistuu lohkojen välissä sijaitsevien ohjelmoitavien kytkimien avulla. Modernit FPGA-piirit sisältävät ohjelmoitavien CLB-lohkojen lisäksi myös lohkoja, joita voidaan käyttää suorittamaan tehokkaasti vain jotain tiettyä ennalta määritettyä funktiota. Yleisesti käytössä olevia tällaisia "hard block"-lohkoja ovat esimerkiksi digitaaliseen signaalinkäsittelyyn vaadittua laskuoperaatiota suorittavat DSP-lohkot (engl. Digital Signal Processing), laskentaan lisämuistia tarjoavat BRAM-muistilohkot (engl. Block Random Access Memory) sekä yksinkertaisten aritmeettisten operaatioiden suorittamiseen erikoistuneet lohkot [5][6].

## 2.2 FPGA-piirin ohjelmoitavuus

Modernit digitaalipiirit yleensä suunnitellaan käyttämällä ohjelmoitavia mikropiirejä kuten FPGA-piirejä, jotka ohjelmoidaan toteuttamaan haluttua toiminallisuutta käyttämällä laitteistonkuvauskieliä (HDL, engl. hardware description language). Laitteistonkuvauskielillä piirin käyttäytymistä pyritään kuvaamaan ohjelmakoodilla ja tätä hyödyntäen piirin toimintaa ja käyttäytymistä voidaan helposti testata simuloimalla suunniteltua HDL-koodia. Varsinainen piiri ohjelmoidaan syntesoimalla laitteistonkuvauskieli synteesityökalun avulla FPGA-piirin ohjelmoitaville lohkoille, jotka edelleen toteuttavat toiminnallisuuden porttitasolla. Laitteistonkuvauskielistä yleisesti suosituimmiksi katsotaan VHDL (engl. Very High Speed Integrated Circuit Hardware Description Language) sekä Verilog. [7]

Laitteistonkuvauskielien, kuten VHDL ja Verilog, sijasta FPGA-piirejä voidaan ohjelmoida myös niin kutsutun korkean tason synteesin avulla, jossa piirien kuvaamiseen käytetään esimerkiksi C- ja C++-ohjelmointikieliä. Tällä tavoin piirejä voidaan kuvata entistä korkeammalla abstraktiotasolla ja piirien ohjelmointi onnistuu yleisesti hyvin käytetyillä ohjelmointikielellä, eikä logiikkapiirien suunnittelijoiden tarvitse opetella HDL-kieliä ainoastaan FPGA-piirien suunnittelua varten. [8]

## 2.3 FPGA-piirien käyttökohteet

FPGA-piireillä on monia hyödyllisiä ominaisuuksia, kuten uudelleenohjelmoitavuus ja mahdollisuus tehokkaaseen laskentaan suorittamalla laskentaoperaatioita rinnakkain. Ominaisuuksiensa, erityisesti joustavuutensa, vuoksi FPGA-piirit ovatkin houkuttelevia vaihtoehtoja usean eri alan sovelluskohteissa [9].

Digitaalisessa signaalinkäsittelyssä nousee arvoonsa erityisesti mahdollisuus suorittaa useita laskutoimituksia samanaikaisesti rinnakkain. Hyödyntämällä FPGA-piirien signaa-

linkäsittelyyn tarkoitettuja DSP-lohkoja, voidaan nopeuttaa huomattavasti vaativaan las- kutoimitukseen kuluva suoritusaikaa verrattuna vastaavaan toteutukseen tavallisella mikroprosessorilla [9][10]. FPGA-piirien avulla voidaan myös toteuttaa tehokkaasti lait- teistotasolla suoritettavia algoritmeja digitaalisen kuvan- ja videonkäsittelyn sovelluk- sissa, kuten piirteiden erottamisessa tai liikkeentunnistuksessa [11].

Nykyään FPGA-piirejä usein myös integroidaan yhteen tavanomaisen keskussyksikön kanssa SoC (engl. System On Chip) FPGA-piireiksi. Näin saavutetaan piirirakenne, joka vie vähemmän tilaa, kuluttaa vähemmän energiaa ja jonka kaistanleveys on suurempi verrattuna sellaiseen piiriin, jossa prosessori ja FPGA-piiri toimivat erillisinä komponent- teinaan. Esimerkkinä tällaisesta SoC FPGA -piiristä on Xilinxin Zynq 7000, jossa integ- roituna ovat ARM Cortex-A9-prosessori sekä joko Artix-7-FPGA- tai Kintex-7-FPGA-piiri [12]. Muita mainitsemisen arvoisia FPGA-piirien sovelluskohteita ovat esimerkiksi bioin- formatiikka, lääketieteen sovellukset, kryptografia, tietoliikennetekniikka, sovellukset so- tilaskäytössä sekä eri sovellusten, erityisesti tekoälyssä ja koneoppimisessa käytettävät, laitteistokiihdyttimet [9].

## 3. KONEOPPIMINEN

Tässä luvussa esitellään koneoppimisen teoreettinen tausta, perehdytään nykyaikaisessa koneoppimisessa monessa sovelluksessa käytettäviin neuroverkkoihin sekä esitellään muutamia koneoppimisen sovelluskohteita.

### 3.1 Koneoppimisalgoritmit

Modernit digitaaliset laitteet ja palvelut ovat riippuvaisia datasta. Erilaisten laitteiden synnyttämä datamäärä on valtava ja tästä datan yltäkylläisyydestä käytetäänkin nimitystä ”big data”. Data ei kuitenkaan ole hyödyllistä, mikäli siitä ei saada mitään informaatiota irti. Tässä tärkeä työkalu on tekoälyn osa-alue koneoppiminen, jonka avulla informaation löytäminen datamassasta on hyvin tehokasta. [13]

Koneoppimisalgoritmit ”oppivat” muokkaamalla omaa käyttäytymistään niille syötetyn datan perusteella niin, että algoritmin ulostulo vastaa haluttua. Sitä, miten koneoppimisalgoritmit oppivat, voidaan käyttää jakamaan koneoppimisalgoritmit karkeasti neljään eri osa-alueeseen. [1]

Ohjatussa oppimisessa (engl. supervised learning) algoritmin opettamiseen käytettävälle datalle on etukäteen määritetty haluttu ulostulo, joka koneoppimisalgoritmin on opittava tuottamaan sille syötetyn datan pohjalta [13]. Tällöin koneoppimisalgoritmi pyrkii oppiesaan muokkaamaan mallinsa parametrejä siten, että sille syötetyn datan ulostulo vastaa mahdollisimman pienellä virhemarginaalilla ennalta määritettyä ulostuloa [1].

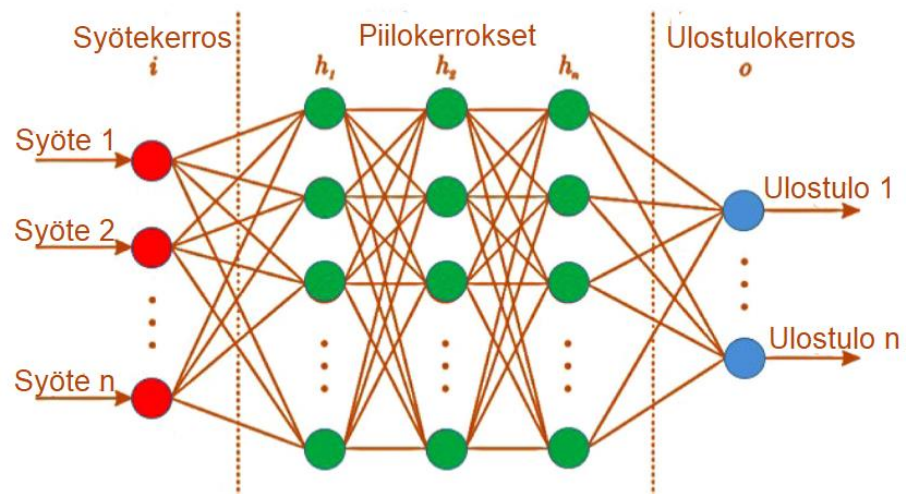
Ohjaamattomassa oppimisessa (engl. unsupervised learning) sen sijaan algoritmi käsittelee ei-luokiteltua dataa ilman ihmisen avustusta ja ohjausta [13]. Ohjaamattoman oppimisen hyöty on se, että toisinaan suurissa datamäärissä on sellaista informaatiota, jota ihminen ei kykene huomaamaan, kuten dataan liittyviä trendejä tai rakenteita [1][13]. Toisaalta ohjaamattomassa oppimisessa on se huono puoli, että koneoppimisalgoritmi voi tuottaa ulostulonaan jotain sellaista informaatioita, mikä ei ole millään tapaa hyödynnettävissä. Koneoppimisessa käytetään myös ohjatun ja ohjaamattoman oppimisen välimuotoa puoliohjattu oppiminen (engl. semi-supervised learning), jota hyödyntävät algoritmit toimivat sekä ohjatusti että ohjaamattomasti, hyödyntäen molempien oppimistapojen ominaisuuksia. [1]

Kolmen edellä esitetyn lisäksi yleisesti käytetty oppimistyyli on palkitsemiseen ja rangaistukseen perustuva oppiminen (engl. reinforcement learning), jossa algoritmi tekee

päätöksiä edellisten päätösten perusteella siten, että se pyrkii saamaan oikeista ratkaisuista saatavien ”palkintojen” määrän mahdollisimman korkeaksi. Tehdessään väärän päätöksen algoritmia rangaistaan ”palkintoja” vähentämällä. [1] Palkitsemiseen ja rangaistukseen perustuvaa oppimista käytetään esimerkiksi robottien ja itseajavien ajoneuvojen koneoppimismallien kouluttamiseksi [13].

### 3.2 Neuroverkot ja niiden rakenne

Neuroverkko (engl. artificial neural network) on koneoppimisalgoritmi, joka muistuttaa peruspiireiltään hyvin paljon ihmisaivojen rakennetta ja toimintaa. Rakenteen keskiössä ovat neuronit, jotka ovat yhteydessä toisiinsa. Algoritmi pyrkii näiden yhteyksien avulla oppia tuottamaan entistä tarkempia tuloksia esimerkiksi luokittelusovelluksissa. [14] Neuroverkkoja käytetäänkin, koska niiden avulla saadaan perinteisiä koneoppimismalleja tehokkaampia malleja erityisesti sellaisiin sovelluksiin, joissa käsiteltävät datamäärät ovat suuria [13]. Neuroverkkojen perusrakenne on esitetty kuvassa 2.



**Kuva 2.** Neuroverkon rakenne (mukaien lähteestä [14]).

Neuroverkot koostuvat useammasta kerroksesta ja yksinkertaisimmillaan ne sisältävät syötekerroksen (engl. input layer), useimmiten ainakin yhden piilokerroksen (engl. hidden layer) ja ulostulokerroksen (engl. output layer). Syötekerroksen neuronit on kytketty kukin ensimmäisen piilokerroksen neuroneihin ja niiden välisten yhteyden voimakkuuden määrittelee yhteyttä vastaavan painon suuruus. Kussakin neuronissa summataan painoilla kerrotut syötearvot, sekä mahdolliset bias-arvot, ja tuotetaan ulostulo syötteeksi

seuraavan kerroksen neuroneille aktivaatiofunktion avulla, mikäli summauksen tulos ylittää halutun kynnyksarvon. Yleensä aktivaatiofunktiona käytetään jotain epälineaarista funktiota kuten sigmoidaalista funktiota [14], jonka avulla neuroverkkoihin saatu epälineaarisuus parantaa algoritmin oppimista erityisesti monimutkaisimmissa ongelmissa. Tällainen toimintaketju jatkuu ulostulokerrokseen saakka, joka tuottaa lopullisen neuroverkon ulostulon [14]. Neuroverkot lopulta oppivat sisään syötetystä datasta säätelämällä kerroksiensa paino- ja bias-arvoja siten, että niiden tuottaman ulostulon virhe olisi mahdollisimman pieni todelliseen ulostuloon verrattuna [14]. Käytännössä neuroverkkoja on monenlaisia ja ne poikkeavat toisistaan hieman joko rakenteeltaan tai toiminnaltaan. Eri-laisia neuroverkkorakenteita ovat esimerkiksi konvoluutioneuroverkot, takaisinkytketyt neuroverkot (RNN, engl. recurrent neural network) ja moniasteperseptronit (MLP, multi-layer perceptron) [13][14].

### 3.3 Koneoppimisen sovelluskohteet

Koneoppimisalgoritmien kyky kehittyä kokemuksensa perusteella tekee niistä hyödyntämiskelpoisia monella eri sektorilla. Erityisesti koneoppiminen nousee arvoonsa sovelluksissa, joissa pyritään automatisoimaan järjestelmä suuren datamäärän pohjalta. Tällaisten sovellusten määrä on kiihtyvässä kasvussa ja näin ollen on myös luontevaa, että myös koneoppimisaiheisten tieteellisten julkaisujen määrä vuosittain kasvaa kovaa vauhtia. [13][1]

Tekoälyyn ja koneoppimiseen ehkä yleisimmin yhdistetyt sovelluskohteet ovat sellaisia, jossa järjestelmä reagoi ihmisaistien tavoin ulkoisiin signaaleihin. Yksi tällainen sovellus on koneoppimisen hyödyntäminen kuvantunnistuksessa, jossa järjestelmä opetetaan tunnistamaan digitaalisista kuvista kohteita kuten ihmiskasvoja tai kirjaimia. Myös koneoppimisen puheentunnistussovellukset ovat suosittuja esimerkiksi älypuhelimien puheella ohjattavissa assistenteissa kuten Google Assistant ja Siri. [13]

Lääketieteen sovellukset ovat yksi suosituimmista koneoppimisen tutkimuskohteista. Koneoppimismalleja voidaan käyttää esimerkiksi tautien ennaltaehkäisyyn ja niiden löytämiseen sekä tukemaan terveydenhuollon päätöksissä [1]. Koneoppimista on myös hyödynnetty COVID-19-pandemiassa monella tapaa, kuten taudin riskiryhmiin kuuluvien henkilöiden määrittelyssä, pandemian aiheuttaneen viruksen alkuperän selvittämisessä sekä taudin diagnosoinnissa ja hoitamisessa. [13][14]

Koneoppimista voidaan hyödyntää myös kyberturvallisuudessa, jonka rooli tulevaisuudessa nousee entisestään esimerkiksi IoT-laitteiden (engl. Internet of Things) suosion kasvun myötä, ja koneoppimisen sovelluksiin kyberturvallisuudessa liittyvät tieteelliset

julkaisut ovatkin kovassa nousussa [1]. Kyberturvallisuuden sovelluksissa koneoppimista voidaan hyödyntää niin yksittäisen ihmisen suojelemiseen kyberturvallisuuden uhilta kuten haittaohjelmilta, mutta myös laajempien kyberhyökkäysten löytämiseen ja niiden torjumiseen [13].

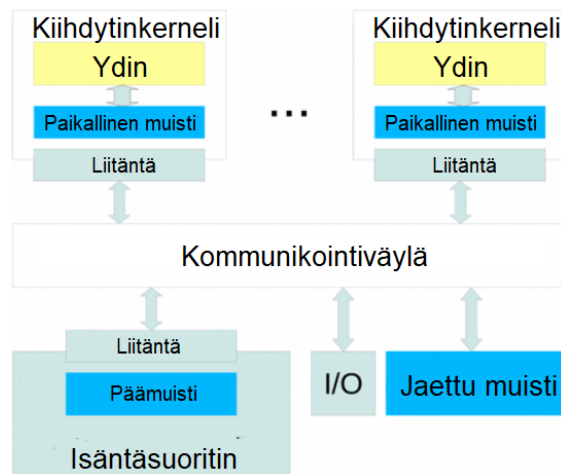
## 4. LAITTEISTOKIIHDYTYKSEN

Tässä luvussa esitellään laitteistokiihdytyksen toimintaperiaate, laitteistokiihdytyksessä yleisesti käytössä olevia kiihdytinteknologioita sekä muutamia tärkeimpiä laitteistokiihdytyksen sovelluksia.

### 4.1 Laitteistokiihdyttimien toimintaperiaate

Vielä 1980- ja 1990-luvulla keskusyksiköiden laskentateho kasvoi kiihtyvää tahtia ja CPU:t kykenivät vastaamaan sen ajan laskennallisiin haasteisiin. Nykyään kuitenkin keskusyksiköiden suoritusnopeuden kehitys on vain muutaman prosentin luokkaa vuosittain [15], minkä vuoksi on syntynyt kasvava tarve kehittää uusia ratkaisuja modernien, laskennallisesti haastavien, ongelmien ratkaisemiseksi. Eräs ratkaisu ongelmaan on käyttää laitteistokiihdyttäjiä. [15]

Laitteistokiihdytintä on tiettyyn laskentatarkoitukseen räätälöity laitteisto, jonka avulla voidaan kiihdyttää järjestelmän laskentaa. Laitteistokiihdytintä ei korvaa keskusyksikköä, vaan toteuttaa laskentaa yhteistyössä sen kanssa. Laitteistokiihdytyksestä hyödyntämällä saavutetaan monia etuja kuten rinnakkaislaskennan tehostuminen, parempi suoritusnopeus, pienempi viive ja parempi energiatehokkuus [16]. Kuvassa 3 on esitetty yksinkertaistettu malli laitteistokiihdyttimien hyödyntävän järjestelmän perusrakenteesta.



**Kuva 3.** Laitteistokiihdyttimien perusrakenne (mukailen lähteestä [17]).

Yksinkertainen laitteistokiihdytinjärjestelmä koostuu isäntäsuorittimesta ja yhdestä tai useammasta lisäsuorittimena toimivasta kiihdytinkernelistä (engl. accelerator kernel),

jotka on toteutettu käytetyllä kiihdytinteknologialla. Isäntäsuoritin ohjaa kerneleitä ja käyttää ohjelman suorittamiseen päämuistia, kun taas lisäsuorittimet käyttävät paikallista muistiaan rinnakkaisuuden parantamiseksi. Usein laitteistokiihdyttimissä käytetään myös jaettua muistia, jonka avulla isäntä- ja lisäsuorittimet jakavat dataa keskenään. [17]

## 4.2 Laitteistokiihdytyksessä käytetyt teknologiat

GPU:ta eli grafiikkasuoritinta käytetään yleisesti grafiikkasovelluksissa, kuten videopeleissä tai videon- ja kuvankäsittelyssä [18]. Grafiikkasuorittimet koostuvat useista ytimiä, jotka eivät yksittäisinä ytiminä kykene vaativiin laskutoimituksiin. Grafiikkasuorittimien suurin hyöty saavutetaankin, kun hyödynnetään näitä ytimiä samanaikaisesti. Käytämällä useita GPU:n ytimiä samanaikaisesti saavutetaan rinnakkaislaskennan avulla suuri suoritusteho [19]. Suuren suoritustehonsa vuoksi grafiikkasuorittimet ovat hyvin suosittuja laitteistokiihdyttiminä ja erityisesti suuren mittakaavan laskentayksiköissä kuten supertietokoneissa [21]. Suurin yksittäinen, sovelluskohteita rajoittava tekijä on grafiikkasuorittimien suuri virrankulutus, minkä vuoksi ne eivät sovellu sovelluksiin, joissa energiankulutus pyritään pitämään vähäisenä [3].

ASIC-piirit ovat tiettyyn käyttötarkoitukseen kustomoituja mikropiirejä [1]. Koska ASIC-piirit suunnitellaan toteuttamaan tiettyä toimintoa, niiden avulla voidaan toteuttaa tehokkaita kiihdyttimiä moniin eri sovelluskohteisiin. ASIC-piirit ovat myös vähän energiaa kulluttavia, mikä tekee niistä hyödyntämiskelpoisia sulautettujen järjestelmien sovelluskohdeissa [20]. ASIC-piirien huonoja puolia ovat esimerkiksi joustamattomuus uudelleenohjelmoitavuuden puutteen vuoksi, sekä paljon aikaa vievä ja kallis tuotanto [3].

FPGA-piirien ohjelmoitavan rakenteen vuoksi ne ovat helposti muokattavissa moneen eri kiihdytinsovellukseen ja näin ollen myös suosittuja laitteistokiihdyttiminä [16]. Kuten ASIC-piirit, ne voidaan ohjelmoida toteuttamaan tehokkaasti tiettyä laskentatarvetta, rinnakkaisuutta hyödyntäen, ja paljon energiatehokkaammin kuin grafiikkasuorittimet. FPGA-piirit kuitenkin tarjoavat ASIC-piirejä enemmän joustavuutta uudelleenohjelmoitavuutensa vuoksi, sillä niiden toiminaalisuutta on mahdollista muovata helposti ja nopeasti piirien valmistusprosessin jälkeenkin. FPGA-piirit ovat myös tuotantokustannuksiltaan halvempia kuin ASIC-piirit, kun tuotanto on kappalemäärältään pientä. FPGA-piirit eivät kuitenkaan laskentateholtaan pärjää joissain kiihdytyssovelluksissa ASIC-piireille ja grafiikkasuorittimille, ja ne vievät enemmän tilaa kuin ASIC-piirit. FPGA-piirejä kuitenkin käytetään usein ASIC-piirien sijasta erityisesti joustavuutensa ja halvemman hintansa vuoksi. [3]

### 4.3 Laitteistokiihdytyksen sovellukset

Tekoäly ja koneoppiminen kehittyvät nopeaa tahtia ja niiden avulla voidaan luoda entistä tarkempia ja nopeampia algoritmeja moniin eri sovelluskohteisiin kuten konenäköön ja hahmontunnistukseen. Suosituin koneoppimismenetelmä monessa sovelluksessa ovat neuroverkot ja erityisesti syväoppimisessa käytetyt monikerroksiset neuroverkot. Algoritmien tarkkuuden kova kehittyminen tuo kuitenkin mukanaan suuren laskentatehon tarpeen, sillä uusimmat neuroverkkomallit vaativat miljardien parametrien käsittelyä ja laskemista. Usein vaativaan laskentaan käytetäänkin erillistä, esimerkiksi FPGA-piireillä toteutettua laitteistokiihdytintä, jolloin saavutetaan neuroverkkojen laskennalle vaadittuja ominaisuuksia kuten matala viive sekä rinnakkaisuuden avulla korkea suorituskyky. Suuri haaste koneoppimisen laitteistokiihdytyksessä on löytää tehokkaita ratkaisuja sovelluksiin, joissa laitteen laskentateho on vaatimaton tai energiankulutus tulisi minimoida. [22][23]

Grafiikkakiihdytin on laitteisto, jota käytetään tietokonegrafiikoiden tuottamiseen. Grafiikkakiihdyttimiä ovat muun muassa tietokoneiden grafiikkasuorittimet [24]. Modernien tietokonegrafiikoiden tuottaminen, esimerkiksi videopeleissä, on laskennallisesti vaativaa, sillä yhden kehyksen (engl. frame) grafiikoiden hahmontamisessa (engl. rendering) laitteisto joutuu käsittelemään miljoonia pikseleitä kerralla sekä tuottamaan kehyksiä usein jopa 60 kappaletta sekunnissa. Grafiikoiden tuottaminen olisi yleiskäyttöisellä CPU:lla liian hidasta ja siksi tietokonegrafiikoiden laskentaan käytetäänkin grafiikkasuorittimia, jotka selviävät vaativasta laskennasta hyödyntämällä ytimiensä avulla saavutettavaa massiivista rinnakkaisuutta [25]. Nykyään realististen grafiikoiden tuottamisessa käytetään usein säteenseurantaa (engl. ray tracing), jossa pyritään mallintamaan valonsäteiden kulkua algoritmin avulla. Säteenseuranta vaatii säteiden kulun mallintamiseksi useiden laskutoimitusten suorittamista samanaikaisesti, minkä vuoksi nykyaikaisissa näyttönohjaimissa on usein oma lohkonsa säteenseurannan laskennalle. [26][25]

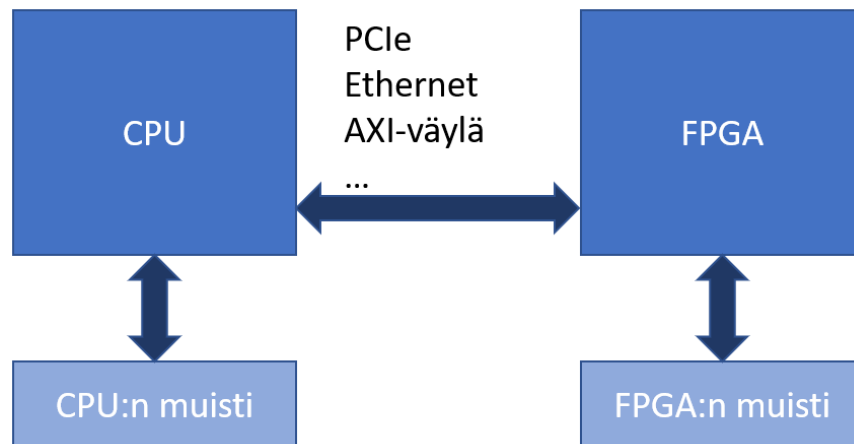
Keskenään kommunikoivat sulautetut järjestelmät, kuten IoT-laitteet, ovat entistä suosittumpia monessa sovelluksessa. IoT-laitteiden tietoturvan toteuttaminen on kuitenkin ongelmallista, sillä ne ovat rajoitettuja suorituskyvyltään ja energiankulutukseltaan, minkä vuoksi salausalgoritmien laskenta laitteen yleissuorittimella on hyvin hidasta. Sen sijaan salausalgoritmin, kuten AES (engl. Advanced encryption standard), laskeminen voidaan toteuttaa erillisellä kiihdytinlaitteistolla, jolloin algoritmin suoritus voidaan toteuttaa paremmalla suoritusteholla ja energiatehokkaammin. Kiihdytinmoduuli voidaan myös toteuttaa helposti muokattavana käyttämällä esimerkiksi FPGA-piirejä, jolloin kiihdyttimeen saadaan joustavuutta toteuttaa monia erilaisia tietoturvasovelluksia. [27][28]

## 5. FPGA-KIIHDYTTIMET KONEOPPIMISESSA

Tässä luvussa käsitellään FPGA-kiihdyttimien yleisrakennetta koneoppimisessa sekä esitellään esimerkki FPGA-pohjaisen koneoppimiskiihdyttimen arkkitehtuurista. Luvussa myös nostetaan esiin FPGA-kiihdyttimien hyötyjä ja ongelmia koneoppimissovelluksissa sekä vertaillaan miten FPGA-kiihdyttimien ominaisuudet poikkeavat muista koneoppimiskiihdytyksessä käytössä olevista teknologioista. Lopuksi tutkitaan FPGA-kiihdyttimien trendejä ja tulevaisuuden näkymiä koneoppimissovelluksissa.

### 5.1 FPGA-pohjaisten koneoppimiskiihdyttimien rakenne

Koneoppimisessa neuroverkkojen ja erityisesti syväoppimisalgoritmien käyttö on entistä suositumpaa, koska niiden avulla voidaan ratkaista monimutkaisiakin datankäsittelyn ongelmia. Tämän vuoksi koneoppimiskiihdyttimien ja erityisesti neuroverkkokiihdyttimiin liittyvän tutkimustyön suosio kasvaa vuosi vuodelta. FPGA-pohjaiset laitteistokiihdyttimet ovat yksi lupaava teknologia neuroverkkojen ja syväoppimisen laskennan kiihdyttämisessä. [29] Kuvassa 4 on esitetty yleinen rakenne FPGA-pohjaiselle neuroverkkokiihdytykselle.

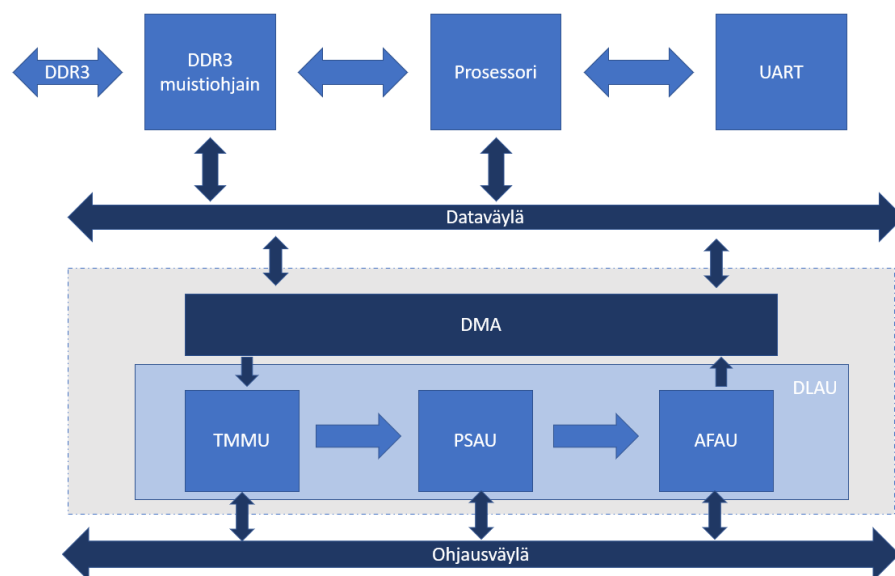


**Kuva 4.** Yleinen FPGA-neuroverkkokiihdyttimen rakenne (mukaillen lähteestä [29])

FPGA-neuroverkkokiihdytin koostuu yksinkertaisimmillaan satojen megahertsien kellotaajuudella toimivan FPGA-piirin lisäksi sitä ohjaavasta keskusyksiköstä, joka vastaa vasti toimii gigahertsien kellotaajuudella. FPGA-piiri ja keskusyksikkö kommunikoivat keskenään niiden välisellä kommunikointiväylällä, joka on toteutettu esimerkiksi PCIe-

(engl. Peripheral Component Interconnect Express) tai Ethernet-teknologialla. Kiihdytynyksikössä on myös usein erillisiä muistikomponentteja, joista erityisesti FPGA:n käyttämä muisti on tärkeässä roolissa, sillä FPGA-piirin sisäisten muistiyksiköiden tallennus-tila ei riitä nykyaikaisten neuroverkkomallien käsittelyyn. Kaupallisten FPGA-piirien sisäinen muistikapasiteetti voi olla suurimmillaan 50 megabittiä, kun neuroverkon parametrit vaativat muistikapasiteettia pienimmilläänkin noin 100 megabittiä. FPGA-piiri voi vaihtoehtoisesti käyttää kiihdyttimen keskussyksikön käyttöön varattua muistia kommunikointiväylää hyödyntäen. [29]

Seuraavaksi esitellään esimerkkiarkkitehtuuri FPGA-pohjaisesta koneoppimiskiihdyttimestä. DLAU (engl. Deep Learning Accelerator Unit) on syväoppimisneuroverkkosten kiihdyttämiseksi suunniteltu FPGA-pohjainen kiihdytynyksikkö [30]. Julkaisussaan [30] L. Gong et al. vertasivat Xilinx XC7Z020-piirille toteutettua DLAU-prototyyppiään 2.3 GHz kellotaajuudella toimivaan Intel Core 2-prosessoriin, ja saavuttivat noin 36-kertaisen nopeutuksen 256x256-kokoisen neuroverkon laskennassa. DLAU hyödyntää FPGA-piirien uudelleenohjelmoitavuutta ja on siten uudelleenskaalattavissa laskemaan eri kokoisia neuroverkkoja mahdollisimman optimoidusti [30]. DLAU-kiihdyttimen perusrakenne on esitelty kuvassa 5.

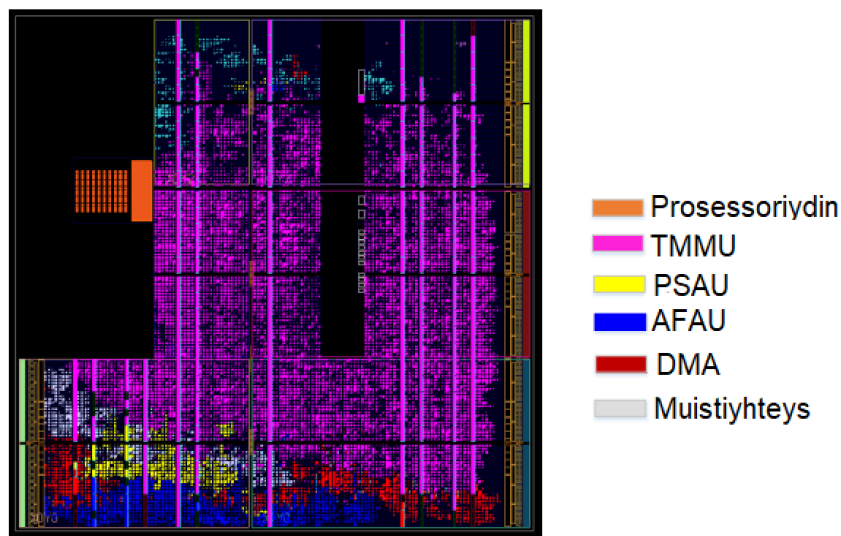


**Kuva 5.** DLAU:n rakenne (mukaillen lähteestä [30])

Itse neuroverkkolaskennan toteuttavan DLAU-lohkon lisäksi kiihdyttimessä on erilaisiin toimintatarkoituksiin alilohkoja. Sulautettu prosessori on järjestelmässä tarkoitettu ohjelmointirajapinnan toteuttamiseksi sekä ohjaamaan DLAU:n toimintaa esimerkiksi käsittelemällä sisään tulevan datan siirtymistä DLAU-yksikön sisäiseen muistiin ja laskennan

lopputuloksen siirtymistä takaisin kiihdyttimen ulkopuoliseen muistiin. DLAU:n kanssa kommunikointiin prosessori käyttää UART-piiriä (engl. Universal Asynchronous Receiver & Transmitter). Kiihdyttimen DDR3-ohjainta (engl. Double Data Rate) ja DMA-moduulia (engl. Direct memory access) tarvitaan lukemaan kiihdyttimen ulkoisesta muistista dataa, kuten neuroverkon painokertoimia, ja myös laskennan lopputuloksen kirjoittamiseen kiihdyttimen ulkopuoliseen muistiin. [30]

DLAU-päälohko koostuu kolmesta liukuhinnoitettusti toisiinsa liitetystä alilohkosta, joista kullakin on oma funktionaalisuutensa neuroverkkomallin laskennassa. TMMU (engl. Tiled Matrix Multiplication Unit) laskee painokertoimien ja sisään tulevan datan välisiä matriisikertolaskuja sekä näiden tulojen osasummia, tallettaen lopputuloksen ulostulopuskuriinsa. PSAU (engl. Part Sum Accumulation Unit) lukee TMMU:lta saatuja tuloksia ja laskee näiden välituloksien osasummia syöttäen ne eteenpäin AFAU:lle (engl. Activation Function Acceleration Unit), joka toteuttaa neuroverkon aktivaatiofunktion käyttäen epälineaarista sigmoidaalista funktiota. Tämän jälkeen lopputulos tallennetaan DMA:n avulla takaisin muistiin [30]. Kuvassa 6 on esitetty DLAU:n fyysinen toteutus FPGA-piirillä.



**Kuva 6.** DLAU-kiihdyttimen toteutus FPGA-piirillä (mukaillen lähteestä [30])

DLAU-kiihdyttimen prototyyppitoteutuksessa eniten pinta-alaa FPGA-piirillä vie TMMU-yksikkö, joka käyttää DLAU:n käyttämistä FPGA-resursseista noin 90 %. TMMU on myös DLAU-prototyypin eniten energiaa kuluttava lohko 189 mW:n kulutuksellaan, kun taas koko kiihdyttimen energiankulutus on 234 mW. [30]

## 5.2 FPGA-pohjaisten koneoppimiskiihdyttimien hyödyt ja ongelmat

Yksi suurimmista syistä, miksi FPGA-piirejä käytetään koneoppimisen kiihdyttämiseksi, on niiden avulla saavutettava rinnakkaisuus [31]. Rinnakkaisuuden avulla FPGA-piirien avulla voidaan laskea rinnakkain samanaikaisesti useita neuroverkkorakenteita, jolloin niiden avulla voidaan toteuttaa erittäin tehokkaita koneoppimiskiihdyttäimiä. FPGA:n rakenteellisten ominaisuuksien takia neuroverkkojen laskennassa vaadittavat laskentayksiköt voidaan toteuttaa liukuhihnoitetusti, mikä soveltuu hyvin neuroverkkojen laskentaan ja tekee laskennasta entistä tehokkaampaa [3]. Tehokkuutta lisää myös laskentaresursien käytön optimointi siten, että FPGA-piireille voidaan toteuttaa vain kiihdyttämiseen tarvittava logiikka sekä jakaa yksittäisiä laskentayksiköitä käytettäväksi monessa eri prosessissa [29].

FPGA-piirien uudelleenohjelmoitavuus tekee niistä joustavia laitteistokiihdyttäimiä neuroverkkosovelluksiin. FPGA-pohjaisten laitteistokiihdyttimien avulla voidaan toteuttaa kiihdytinratkaisuja, jotka eivät sovellu ainoastaan yhdenlaiseen koneoppimissovellukseen, vaan niitä voidaan myös muovata eri käyttökohteisiin sopiviksi. Uudelleenohjelmoitavuuden avulla voidaan esimerkiksi optimoida FPGA-piirin laskenta- ja muistiresursseja. Muokkaamista ja säätöä vaaditaan usein esimerkiksi erikokoisten neuroverkkojen laskennassa [32]. Uudelleenohjelmoitavuus myös tarjoaa kiihdyttimen kehitysprosessissa mahdollisuuden arvioida lyhyessä ajassa kiihdyttimen toimivuutta ja tehdä tarvittavia korjauksia nopeasti [33].

Suuri etu FPGA-koneoppimiskiihdyttimissä on myös FPGA-piirien alhainen virrankulutus. Vähäisen virrankulutuksensa ja tehokkuutensa vuoksi niiden avulla voidaan toteuttaa koneoppimiskiihdyttäimiä sellaisiin sovelluksiin, joissa energiankulutus halutaan minimoida. Tällaisia FPGA-piireille sopivia vähän energiaa kuluttavia sovelluskohteita ovat esimerkiksi akuilla toimivat laitteet, kuten mobiililaitteet, sulautetut järjestelmät sekä pilvipalveluiden palvelimet. [2][33]

FPGA-pohjaisten koneoppimiskiihdyttimien ongelmaksi nousee kuitenkin koneoppimisessa vaadittava suuri laskenta- ja muistiresurssien määrä, sillä neuroverkot kasvavat kooltaan jatkuvasti [33]. Nykyaikaiset neuroverkot sisältävät kymmeniä miljoonia parametrejä ja niiden tallettaminen kaupallisten FPGA-piirien tallennuskapasiteetiltaan pieneen sisäiseen muistiin ei ole mahdollista. Näin ollen FPGA-koneoppimiskiihdyttimissä joudutaan usein turvautumaan ulkoiseen muistiin, mikä heikentää kiihdyttimen suorituskykyä. Kymmenien miljoonien parametrien käsittely on myös laskennallisesti hyvin vaa-

tivaa ja paljon laskentaresursseja vievää, mikä luo haasteita FPGA-piirin rajallisten laskentaresurssien kanssa. Eräs ratkaisu laskentaresurssipulaan on käyttää samoja laskentayksiköitä useassa eri laskentaoperaatiossa ja hyödyntää esimerkiksi osasummaa. [2]

FPGA-piirien uudelleenohjelmoitavuuden avulla voidaan kehittää joustavia koneoppimiskiihdyttäjiä, mutta uudelleenohjelmoitavuudessakin on ongelmansa. FPGA-kiihdyttimien ohjelmoitavuudessa ongelmallista on erityisesti niiden toiminnallisuuden määrittelyssä edelleen useasti käytettävät laitteistonkuvauskielet, joiden suosio on pientä CPU-pohjaisissa kiihdyttimissä käytettävien korkeamman abstraktiotason kieliin verrattuna. [31]

### 5.3 FPGA-pohjaisten koneoppimiskiihdyttimien vertailu muihin kiihdytinteknologioihin

GPU:t, eli grafiikkasuorittimet, ovat hyvin laajasti käytettyjä laitteistokiihdyttäjiä koneoppimisessa, erityisesti monimutkaisten neuroverkkojen ja syväoppimismallien kiihdyttämisessä. GPU koostuu rakenteeltaan useista rinnakkaisista prosessoriytimistä, minkä vuoksi ne soveltuvat erittäin hyvin rinnakkaisuutta paljon vaativiin, laskennallisesti haastaviin, laskentasovelluksiin. Grafiikkasuorittimien sovelluskohteita ovatkin syväoppimista hyödyntävät sovelluskohteet kuten kasvojentunnistus, tiedonlouhinta ja itseajavat ajoneuvot. Tehokkaita moderneja GPU-pohjaisia koneoppimiskiihdyttäjiä ovat esimerkiksi Nvidian kehittämät Nvidia A100 ja Nvidia V100. [3]

GPU-kiihdyttimien yksi merkittävimpiä eroja FPGA-pohjaisiin koneoppimiskiihdyttäjiin on se, että GPU-koneoppimiskiihdyttimet kuluttavat hyvin paljon virtaa. Yleensä GPU-pohjaiset kiihdyttimet koostuvat suuresta määrästä aritmeettisloogisista yksiköistä (engl. Arithmetic Logic Unit), jotka eivät voi kommunikoida suoraan keskenään. Tästä syystä aritmeettisloogisten yksiköiden on kommunikoitava niille yhteisen DRAM- muistin avulla, mikä aiheuttaa laskennan aikana paljon muistioperaatioita, jotka edelleen kuluttavat virtaa. Toinen merkittävä tekijä virrankulutukselle on GPU-kiihdyttimien käyttämä liukulaskenta, joka on laskennallisesti vaativaa ja näin ollen myös ongelma virrankulutuksen kannalta [34]. GPU-kiihdyttimien virrankulutus nouseekin jopa satoihin watteihin, mikä tekee niiden virrankulutuksesta kymmeniä kertoja suuremman kuin FPGA-pohjaisilla kiihdyttimillä [35][3]. Suuren virrankulutuksensa vuoksi GPU-kiihdyttimet soveltuvat näin ollen FPGA-kiihdyttäjiä huonommin sovelluksiin, joissa virrankulutuksen minimointi on tärkeää, kuten datakeskukset ja sulautetut järjestelmät [30]. FPGA-kiihdyttimet myös soveltuvat GPU-pohjaisia kiihdyttäjiä paremmin sellaisiin koneoppimissovelluksiin,

joissa mallia, esimerkiksi neuroverkkoa, on muokattava useasti esimerkiksi parametreiltaan optimoinnin vuoksi [36].

Mikäli koneoppimissovelluksessa kuitenkin käsitellään hyvin monimutkaisia, laskennallisesti vaativia neuroverkkoja, saavutetaan grafiikkasuorittimilla paljon nopeampi ja suoritusteholtaan parempi kiihdytinvaihtoehto. FPGA-pohjaisten laitteistokiihdyttimien ongelma on niiden vähäiset laskenta- ja muistiresurssit sekä alhainen kaistanleveys muistioperaatioissa, minkä vuoksi ne eivät sovellu kaikista monimutkaisimpien neuroverkkojen kiihdytykseen [3]. GPU-kiihdyttimillä sen sijaan sekä laskenta- ja muistiresurssit että käytettävä kaistanleveys ovat mittaluokkaa suurempia. GPU-kiihdyttimillä saavutetaan myös suurempi kelloaajuus ja rinnakkaisuus, jolloin niiden suoritustehoa mitataan yleensä TOPS:eissa (engl. Tera Operations Per Second) kun taas FPGA-kiihdyttimillä suoritusteho on tyypillisesti GOPS (engl. Giga Operations Per Second) mittaluokassa [34][3].

FPGA- ja GPU-kiihdyttimien lisäksi koneoppimisen kiihdyttämisessä käytetään usein myös ASIC-piirejä. Kuten FPGA-kiihdyttimillä, ASIC-pohjaisten kiihdyttimien virrankulutus ovat hyvin vähäistä, minkä vuoksi niitä hyödynnetään vähän energiaa kuluttavissa sovelluksissa. FPGA-kiihdyttimien kanssa yhtäläistä on myös se, että ASIC-kiihdyttimet eivät ole yleiskäyttöisiä suorittimia kuten GPU:t, vaan ne suunnitellaan toteuttamaan tiettyä laskentaoperaatiota. Tästä syystä ASIC-piireillä saadaan aikaan hyvin tehokkaita, tiettyyn koneoppimiskiihdytykseen suunniteltuja kiihdyttimiä [3]. Eräs tehokas moderni ASIC-pohjainen koneoppimiskiihdytin on Googlen kehittämä TPU-kiihdytin (engl. Tensor Processing Unit). TPU on tehokas, vähän energiaa kuluttava ja halpa koneoppimisen kiihdyttämiseksi tarkoitettu kiihdytin, jota käytetään esimerkiksi Googlen datakeskuksissa hakukoneen hakutuloksien löytämisen nopeuttamiseksi. Nopeus saavutetaan muun muassa siksi, että kiihdytin toteuttaa laskentaa pienemmällä tarkkuudella. [36][37]

Nykyaikaiset ASIC-kiihdyttimet ovat yleisesti ottaen nopeampia kuin vastaavat FPGA-kiihdyttimet koneoppimisen kiihdyttämisessä. ASIC-piirien suuri etu on se, että niiden laskentaresurssit saadaan optimoitua FPGA-piirejä paremmin käyttötarkoitukseen sopivaksi, jolloin saavutetaan sekä FPGA-kiihdyttimiä että GPU-kiihdyttimiä paljon suurempi suoritusteho. FPGA-piireissä laskentaresurssien optimointia rajoittaa loogisten yksiköiden välisen reitityksen rajoitteet. [3][37] ASIC-kiihdyttimet myös vievät FPGA-kiihdyttimiä vähemmän virtaa ja näin ollen niiden energiatehokkuus on parempi [3]. Tämän lisäksi ne ovat yksikköhinnaltaan halvempia ja pinta-alaltaan pienempiä [35].

ASIC-piirit kuitenkin häviävät ominaisuuksiltaan FPGA-piireille siinä, että ne eivät ole uudelleenohjelmoitavia. Tämä heikentää merkittävästi niiden joustavuutta ja rajoittaa niiden

käyttämistä sellaisissa sovelluksissa, joissa koneoppimismallin ominaisuuksia, kuten neuroverkon kokoa, on muokattava käyttötarkoitukseen optimiksi. [16] ASIC-pohjaisten kiihdyttimien ongelma on myös se, että niiden tuotekehityksen NRE-kustannukset (engl. Non-recurring engineering) nousevat FPGA-kiihdyttäjiä paljon korkeammaksi. Tämän lisäksi niiden markkinointiprosessi on FPGA-pohjaisia kiihdyttäjiä hitaampaa. ASIC-kiihdyttimien tuotekehityksen hitauden vuoksi niillä on vaikeuksia pysyä mukana hyvin nopeasti kehittyvillä koneoppimis- ja neuroverkkokiihdyttimien markkinoilla. [16][34]

**Taulukko 1.** ASIC- ja GPU-koneoppimiskiihdyttimien vertailu FPGA-kiihdyttimiin

Kiihdytin- teknologia	Nopeus ja suoritus-teho	Laskenta- ja muistiresurssit	Tehon- kulutus	Uudelleen- ohjelmoitavuus
ASIC	Nopeampi ja suoritus-teholtaan parempi teknologia	Kiihdyttimen laskentaresurssit optimoitavissa paremmin kiihdytinkäyttöön	Parempi energiatehokkuus ja pienempi virrankulutus	Ei uudelleenohjelmoitavissa
GPU	Suoritus-teholtaan parempi, ja soveltuu suurempien mallien kiihdyttämiseen	Kiihdyttimellä paljon suuremmat laskenta- ja muistiresurssit	Paljon suurempi virrankulutus	Ei yhtä joustavasti uudelleenohjelmoitavissa

Taulukossa 1 on esitetty yhteenvetona ASIC- ja GPU-koneoppimiskiihdyttimien tärkeimpien ominaisuuksien vertailu FPGA-pohjaisiin koneoppimiskiihdyttimiin. Taulukosta huomataan, että FPGA-kiihdyttimien suurin vahvuus muihin teknologioihin verrattuna on niiden helpon uudelleenohjelmoitavuuden tarjoama joustavuus. Heikkouksia FPGA-kiihdyttimillä sen sijaan ovat esimerkiksi alhaisempi suoritus-teho sekä vähäiset laskenta- ja muistiresurssit.

## 5.4 FPGA-pohjaisten koneoppimiskiihdyttimien merkitys tulevaisuudessa

Tulevaisuudessa käsiteltävän datan määrä tulee kasvamaan entisestään. Datan käsitteilyssä ja siitä informaation löytämisessä ihmisiä ylivermaisempia toimijoita ovat tekoälyalgoritmit ja erityisesti neuroverkot, minkä vuoksi niiden merkitys tulee olemaan myös tulevaisuudessa suuri [38]. Tekoäly osa-alueena on kuitenkin vielä alkutekijöissään, joten tulevaisuudessa koneoppimis- ja neuroverkkomallit tulevat olemaan entistä monimutkaisia, mikä kasvattaa tarvetta entistä tehokkaammille ja optimoidummille laskentalaitteistoille kuten laitteistokiihdyttimille [38][3].

FPGA-piirien markkinat ovat kovassa kasvussa ja tämä kasvukäyrä tulee jatkumaan myös tulevaisuudessa, mikä tulee näkymään myös FPGA-kiihdyttimien suosiossa [9].

FPGA-pohjaiset laitteistokiihdyttimet nousevat tulevaisuudessa suureen rooliin erityisesti sellaisissa koneoppimisen sovelluksissa, joissa energiankulutus tulee minimoida, kuten IoT-laitteet ja sulautetut järjestelmät, joiden määrä tulevaisuudessa tulee kasvamaan. FPGA-piireihin pohjautuvat laitteistokiihdyttimet tulevat myös olemaan merkittävässä roolissa reaaliaikaista prosessointia vaativissa sovelluksissa, kuten itseajavissa autoissa, joiden laskentayksiköiden on käsiteltävä valtava määrä dataa mahdollisimman pienellä viiveellä. [39]

Yksi merkittävä koneoppimisen sovelluskohde FPGA-kiihdyttimille tulee olemaan pilvipalveluiden laskennan kiihdyttäminen, jossa vaaditaan erityisesti FPGA-piireille ominaisia piirteitä, kuten vähäinen energiankulutus ja pieni viive. FPGA-kiihdyttimien käyttö pilvipalveluiden kiihdyttämisessä ei ole vielä kovin laajassa suosiossa, mutta isoista yrityksistä esimerkiksi Microsoft käyttää FPGA-kiihdyttämiä jopa miljoonan palvelimen laskennan kiihdyttämiseksi Azure- pilvipalvelussaan. [3][39]

Microsoft on myös mukana Intelin kanssa Brainwave- nimisessä projektissa, jossa Microsoftin erilaisten palveluiden, kuten Bing-hakukone, nopeutta pyritään lisäämään käyttämällä syväneuroverkkoja ja niiden kiihdyttämiseen Intelin kehittämällä FPGA-kiihdyttimillä kuten Intel Stratix 10 FPGA. Laitteistokiihdyttimen ydin on FPGA-piirille toteutettu Brainwave NPU-laskentayksikkö (engl. neural processing unit), joka on suunniteltu kiihdyttämään syväneuroverkkojen laskentaa. Itse NPU koostuu pääosan matriisilaskennasta toteuttavasta NFU:sta (engl. neural functional unit) sekä sen sisällä olevista pienemmistä MFU-laskentayksiköistä (engl. multifunctional unit), jotka muun muassa toteuttavat neuroverkon aktivaatiofunktion. NPU-laskentayksiköiden käytöllä Microsoftin palveluiden laskennan kiihdyttämisessä on saavutettu lupaavia tuloksia. Esimerkiksi Bing-hakupalvelun älykkäässä hakutoiminnossa saavutettiin kymmen kertaa pienempi viive FPGA-pohjaisella NPU:lla kuin vastaavan hakutoiminnon laskeminen CPU-pohjaisella ratkaisulla, vaikka NPU:n käyttämä neuroverkkomalli oli kymmenen kertaa suurempi. [40]

FPGA-kiihdyttimien tulevaisuus monissa koneoppimisen sovelluksissa on lupaava. FPGA-pohjaisten koneoppimiskiihdyttimien tutkimuksessa ja kehityksessä on kuitenkin ratkaistavana ongelmia, jotka vielä tällä hetkellä rajoittavat niiden käyttöä entistä laajemmin koneoppimissovelluksissa. Yksi tällainen tutkittava ongelma on FPGA-piirien vähäiset resurssit niin laskenta- kuin muistiyksiköiden koossa. Neuroverkkojen koko ja monimutkaisuus tulevaisuudessa kasvaa entisestään, jolloin esimerkiksi aktivaatiofunktioiden ja matriisilaskennan laskeminen FPGA-kiihdyttimillä on nykyisillä resursseilla haastavaa. Muistin suhteen ongelman ratkaisuksi on esitetty ulkoisen muistin käyttöä siten, että muistiväylän kaistanleveys olisi mahdollisimman optimoitu FPGA-piireille. Ongelmallista

on myös FPGA-kiihdyttimien ohjelmointi ja helppokäyttöisen ohjelmointialustan, kuten GPU-kiihdyttimien ohjelmointiin käytetty NVIDIA CUDA (engl. Compute Unified Device Architecture), kehittäminen, jonka avulla voitaisiin siirtää sovelluksia nopeasti FPGA-piirille. [3][29]

## 6. YHTEENVETO

FPGA-piirit ovat uudelleenohjelmoitavia mikropiirejä, jotka koostuvat matriisirakenteen muodostavista, toisiinsa liitetyistä, ohjelmoitavista logiikkalohkoista. Nykyaikaiset FPGA-piirit sisältävät myös useita tiettyyn käyttötarkoitukseen, kuten signaalinkäsittelyyn, tarkoitettuja lohkoja. FPGA-piirit ohjelmoidaan käyttäen laitteistokuvauskieliä ja niiden uudelleenohjelmoitavuuden tarjoama joustavuus tekee niistä hyödyntämiskelpoisia monessa eri sovelluksessa. FPGA-piireillä voidaan myös tehokkaasti suorittaa rinnakkaislaskentaa, mikä tekee niistä suosittuja digitaalitekniikan sovelluksissa.

Koneoppimisen ja erityisesti neuroverkkojen suosio on suuressa kasvussa johtuen tarpeesta käsitellä massiivisia datamääriä tehokkaasti. Koneoppimiseen liittyvät tutkimusjulkaisut kasvavat vuosi vuodelta tuoden tarjolle entistä tehokkaampia, mutta toisaalta myös monimutkaisempia ja suurempia, koneoppimisalgoritmeja. Kasvavan kokonsa ja monimutkaisuutensa vuoksi koneoppimisalgoritmit vaativat entistä tehokkaampia laskentayksiköitä ja yleensä niiden laskentaa suoritetaan erityisesti laskentaan tarkoitettulla laitteistokiihdyttimellä. Laitteistokiihdyttimet ovat yleensä keskusyksikön kanssa yhteistyössä toimivia laskentayksiköitä. Laitteistokiihdyttämiä käyttämällä laskentaa voidaan suorittaa rinnakkain ja saavuttaa muun muassa korkeampi suoritusteho, pienempi viive ja parempi energiatehokkuus kuin toteuttamalla laskenta yleiskäyttöisellä suorittimella. Laitteistokiihdyttimissä usein käytettyjä teknologioita ovat GPU:t, ASIC-piirit ja FPGA-piirit.

FPGA-piireihin pohjautuvat koneoppimiskiihdyttimet koostuvat perusrakenteeltaan laskennan toteuttavasta FPGA-piiristä, sitä ohjaavasta keskusyksiköstä, näiden välisestä kommunikointiväylästä sekä muistiyksiköistä. FPGA-piireistä on kehitetty monia toisistaan hieman arkkitehtuuriltaan ja ominaisuuksiltaan poikkeavia koneoppimiskiihdyttämiä. Esimerkki FPGA-piiriin pohjautuvasta koneoppimiskiihdyttimestä on DLAU-laitteistokiihdytin, joka hyödyntää FPGA-piirien uudelleenohjelmoitavuutta tarjoten käyttäjälleen helpon tavan optimoida kiihdytin eri kokoisille neuroverkoille. DLAU-kiihdyttimessä FPGA-piirille on toteutettu usea liukuhinamaisesti toisiinsa kytketty alilohko, joista jokainen toteuttaa omaa laskentatarkoitustaan.

FPGA-piirit tarjoavat useita ominaisuuksia, joita voidaan hyödyntää koneoppimiskiihdytyksessä. FPGA-kiihdyttimien tarjoama suuri rinnakkaisuus mahdollistaa tehokkaan laskentayksikön valmistamisen neuroverkkorakenteiden laskentaan. Uudelleenohjelmoitavuutensa vuoksi ne voidaan myös helposti muokata eri käyttötarkoituksiin sopiviksi.

Suuri etu FPGA-kiihdyttimillä on myös se, että niiden virrankulutus on vähäistä, minkä vuoksi ne soveltuvat hyvin sellaisiin koneoppimissovelluksiin, joissa energiankulutus tulee minimoida, kuten sulautetut järjestelmät tai pilvipalvelut. Vertaillen FPGA-kiihdyttimien ominaisuuksia muihin koneoppimiskiihdytyksessä käytettäviin teknologioihin, GPU- ja ASIC-kiihdyttimiin, erottuvat FPGA-pohjaiset kiihdyttimet erityisesti pienellä virrankulutuksellaan ja suurella joustavuudellaan. FPGA-kiihdyttimien suurimpia ongelmia sen sijaan ovat esimerkiksi niiden vähäiset laskenta- ja muistiresurssit sekä monista korkeamman abstraktiotason ohjelmointikielistä poikkeavien laitteistonkuvauskielien käyttö piirien ohjelmoinnissa ja suunnittelussa.

Tulevaisuudessa FPGA-kiihdyttimet tulevat olemaan suosittuja erityisesti sellaisissa koneoppimissovelluksissa, joissa vaaditaan joustavia, reaaliaikaisia ja vähän energiaa kulluttavia kiihdyttäjiä. FPGA-koneoppimiskiihdyttimien tulevaisuuden potentiaalisia sovel- luskohhteita ovat esimerkiksi koneoppimisalgoritmien kiihdytys sulautettujen järjestelmien sovelluksissa, itseajavien ajoneuvojen datankäsittelyssä ja pilvipalvelimien laskennassa, joissa jo esimerkiksi Microsoft käyttää FPGA-piirejä pilvipalvelimiensa laskennan kiihdyttämiseksi.

# LÄHTEET

- [1] R. Marini, R. Pugliese, S. Regondi, Machine learning-based approach: global trends, research directions, and regulatory standpoints, *Data Science and Management*, 2021, Vol.4, pp.19-29
- [2] A. El-Maleh, S.M. Sait, A. Shawahna, FPGA-Based Accelerators of Deep Learning Networks for Learning and Classification: A Review, *IEEE access*, 2019, Vol.7, pp.7823-7859
- [3] S. Boppu, L.R. Cenkeramaddi, P. Dhilleswararao, M.S. Manikandan, Efficient Hardware Architectures for Accelerating Deep Neural Networks: Survey, *IEEE access*, 2022, Vol. 10, pp. 131788-131828
- [4] I. Kuon, R. Tessier, J.Rose, *FPGA Architecture: survey and challenges*, Boston: Now Publishers, 2008
- [5] U. Farooq, Z. Marrakchi, H. Mehrez. *Tree-Based Heterogeneous FPGA Architectures: Application Specific Exploration and Optimization*. 1. Aufl. Vol. 9781461435945. New York, NY: Springer-Verlag, 2012. Web.
- [6] K. Iniewski. *Embedded Systems: Hardware, Design and Implementation*. Somerset: John Wiley & Sons, Incorporated, 2012
- [7] N. Botros, *HDL with Digital Design: VHDL and Verilog*, Mercury Learning & Information, 2015
- [8] P. Coussy, A.Morawiec, *High-Level Synthesis from Algorithm to Digital Circuit*, 1st ed, Springer Dordrecht, 2008
- [9] P. Babu, P. Eswaran, "Reconfigurable FPGA Architectures: A Survey and Applications.", *Journal of the Institution of Engineers (India). Series B, Electrical Engineering, Electronics and telecommunication engineering, Computer engineering* 102, no. 1 (2021): pp. 143–156.
- [10] G. Singh, *Reconfigurable computing: a review of the technology and its architecture*, *IOSR Journal of VLSI and Signal Processing*, 2013, Vol.3 (4), pp.8-14
- [11] I. Bravo-Muñoz, A.Gardel-Vicente, J.L. Lázaro-Galilea, *FPGA and SoC Devices Applied to New Trends in Image/Video and Signal Processing Fields*, *Electronics (Basel)*, 2017, Vol.6 (2), pp.25
- [12] Intel, *What is an SoC FPGA?* (viitattu: 23.1.2023), saatavissa: <https://www.intel.com/content/dam/support/us/en/programmable/support-resources/bulk-container/pdfs/literature/ab/ab1-soc-fpga.pdf>
- [13] I.H. Sarker, *Machine Learning: Algorithms, Real-World Applications and Research Directions*, *SN Computer Science*, vol. 2, no. 3, 2021, pp. 160–160
- [14] A.Ajith, A.K. Tyagi, *Recurrent Neural Networks : Concepts and Applications*. Edited by Amit Kumar Tyagi and Ajith Abraham, First edition., CRC Press, 2023

- [15] S. Keckler, D. Milojicic, Accelerators, Computer (Long Beach, Calif.), vol. 55, no. 1, 2022, pp. 108–112
- [16] S. Bartolini, M. Mannino, A. Mondelli, B. Peccerillo, A Survey on Hardware Accelerators: Taxonomy, Trends, Challenges, and Perspectives. Journal of Systems Architecture, vol. 129, 2022, p. 102561
- [17] Z. Al-Ars, I. Ashraf, K. Bertels, P. Cuong, Heterogeneous Hardware Accelerators with Hybrid Interconnect: An Automated Design Approach, 2015 International Conference on Advanced Computing and Applications (ACOMP), IEEE, 2015, pp. 59–66
- [18] A. Baobaid, M. Meribout, J.P. Pena, V.K. Tiwari, Hardware Accelerators for Real-Time Face Recognition: A Survey, IEEE Access, vol. 10, 2022, pp. 83723–83739
- [19] S. Jafar, G. Yang, Design Flow of Single Camera Motion Estimation Using GPU Accelerators, 2019 IEEE International Conference on Electro Information Technology (EIT), vol. 2019-, IEEE, 2019, pp. 185–188
- [20] J.R. Azambuja, M. Brandalero, L. Gobatto, M.M. Goncalves, S. Pagliarini, T.D. Perez, G-GPU: A Fully-Automated Generator of GPU-Like ASIC Accelerators, 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), EDAA, 2022, pp. 544–547
- [21] M. Hossain, R. Machupalli, M. Mandal, Review of ASIC Accelerators for Deep Neural Network, Microprocessors and Microsystems, vol. 89, 2022, p. 104441
- [22] S. Bavikadi, A. Dhavlle, A. Ganguly, A. Haridass, H. Hendy, C. Merkel, V.J. Reddi, P.R. Sutradhar, A. Joseph, S.M. Pudukotai Dinakarrao, A Survey on Machine Learning Accelerators and Evolutionary Hardware Platforms, IEEE Design and Test, vol. 39, no. 3, 2022, pp. 91–116
- [23] D. Ghimire, D. Kil, S. Kim, A Survey on Efficient Convolutional Neural Networks and Hardware Acceleration, Electronics (Basel), vol. 11, no. 6, 2022, p. 945
- [24] A.B. Craig. Understanding Augmented Reality Concepts and Applications. 1st edition, Morgan Kaufmann, 2013.
- [25] K. Akeley, S.K. Feiner, J.D. Foley, J.F. Hughes, M. McGuire, D.F. Sklar, A. van Dam. Computer Graphics: Principles and Practice, Third Edition, Addison-Wesley Professional, 2013
- [26] Y. Chen, B. Gao, M. Gong, L. Peng, R. Yin, Ray Tracing on Single FPGA, 2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), IEEE, 2021, pp. 1290–1294
- [27] A. Salman, E. Samy, Efficient Hardware/Software Co-Design of Elliptic-Curve Cryptography for the Internet of Things, 2019 International Conference on Smart Applications, Communications and Networking (SmartNets), IEEE, 2019, pp. 1–6
- [28] C. Babecki, S. Bhunia, R. Karam, S. Paul, Q. Wenchao, An Embedded Memory-Centric Reconfigurable Hardware Accelerator for Security Applications, IEEE Transactions on Computers, vol. 65, no. 10, 2016, pp. 3196–3202

- [29] N.M Philip, N. M. Sivamangai, Review of FPGA-Based Accelerators of Deep Convolutional Neural Networks, 2022 6th International Conference on Devices, Circuits and Systems (ICDCS), IEEE, 2022, pp. 183–189
- [30] L. Gong, C. Wang, X. Li, Y. Xie, Q. Yu, X. Zhou, DLAU: A Scalable Deep Learning Accelerator Unit on FPGA, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 36, no. 3, 2017, pp. 513–517
- [31] H. Chen, C. Wang, H. Wang, X. Zhou, An Overview of FPGA Based Deep Learning Accelerators: Challenges and Opportunities, 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), IEEE, 2019, pp. 1674–1681
- [32] N. Ali, P. Coussy, J.M. Philippe, B. Tain, Exploration and Generation of Efficient FPGA-Based Deep Neural Network Accelerators, The Institute of Electrical and Electronics Engineers, Inc. (IEEE) Conference Proceedings, The Institute of Electrical and Electronics Engineers, Inc. (IEEE), 2021
- [33] M. Bedoui, A. Blaiech, M. Fernandes, K. Khalifa, C. Valderrama, A Survey and Taxonomy of FPGA-Based Deep Learning Accelerators, Journal of Systems Architecture, vol. 98, 2019, pp. 331–345
- [34] L. Baischer, N. Taherinejad, M. Wess, Learning on Hardware: A Tutorial on Neural Network Accelerators and Co-Processors, arXiv.org, 2021, saatavissa (viitattu 16.2.2023) <https://arxiv.org/pdf/2104.09252.pdf>
- [35] B. Bussolino, M. Capra, A. Marchisio, G. Masera, M. Shafique, Hardware and Software Optimizations for Accelerating Deep Neural Networks: Survey of Current Trends, Challenges, and the Road Ahead, IEEE Access, vol. 8, 2020, pp. 225134–225180
- [36] Y. Hu, Y. Liu, Z. Liu, A Survey on Convolutional Neural Network Accelerators: GPU, FPGA and ASIC, 2022 14th International Conference on Computer Research and Development (ICCRD), IEEE, 2022, pp. 100–107
- [37] M. Alsaraf, F. Jasem, A Survey of Neural Network Hardware Accelerators in Machine Learning, Machine Learning and Applications: An International Journal, vol. 8, no. 4, 2021, pp. 11–28
- [38] T. Chen, Z. LI, Y. Wang, T. Zhi, A Survey of Neural Network Accelerators, Frontiers of Computer Science, vol. 11, no. 5, 2017, pp. 746–761
- [39] W. Hwu, S. Patel, Accelerator Architectures -A Ten-Year Retrospective, IEEE MICRO, vol. 38, no. 6, 2018, pp. 56–62
- [40] Chung, Eric, et al, Serving DNNs in Real Time at Datacenter Scale with Project Brainwave, IEEE MICRO, vol. 38, no. 2, 2018, pp. 8–20