

Piikkinerooverkkojen syväoppiminen ja niiden
toteutus neuromorfisissa systeemeissä

LuK-tutkielma
Turun yliopisto
Fysiikka
2026
Yunus Polat
Tarkastaja:
Prof. Petriina Paturi

Turun yliopiston laatujärjestelmän mukaisesti tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck-järjestelmällä

TURUN YLIOPISTO

Fysiikan laitos

Polat, Yunus Pikkineroverkkojen syväoppiminen ja niiden toteutus neuromorfisissa systeemeissä

LuK-tutkielma, 24 s., 3 liites.

Fysiikka

Tammikuu 2026

Neuroverkot on rakennettu neuroneista, jotka ovat kytketty toisiinsa synapsien avulla. Jotta neuroverkot saadaan toimimaan halutulla tavalla ja aktivoitumaan tietyissä tilanteissa, opetetaan niille synaptiset painot. Näitä tarkkojen painojen asettamista kutsutaan syväoppimiseksi. Tätä hyödyntävissä syvissä neuroverkoissa on neuroneita monissa kerroksissa, ja juuri tämä monikerroksinen arkkitehtuuri on tuottanut lupaavia tuloksia koneoppimiseen ja tekoälyyn liittyvissä tutkimuksissa. Neuromorfisten systeemien arkkitehtuuri pohjautuu biologisiin hermostoihin, ja niiden avulla pyritään luomaan energiatehokas tapa laskea. Nämä systeemit eivät tällä hetkellä käytä piikkineroverkkoja, mutta silti nykyiset neuromorfiset systeemit, sekä analogiset että digitaaliset, ovat olleet lupaavia. Niitä on kuitenkin tutkittu vain yksinkertaisilla tutkimuksilla eivätkä ne kykene tällä hetkellä yhtä haastaviin tehtäviin kuin perinteiset tietokoneet. Tarvitaan yhteistyötä monien eri alojen osajilta, jotta näitä systeemejä saadaan tutkittua lisää, jotta löydetään vähemmän tilaa vievä ja energiatehokkaampi vaihtoehto laskennalle.

Asiasanat: neuromorfinen, syväoppiminen, piikkineroverkko, neuroverkko

Sisällys

Johdanto	1
1 Neuroverkot	2
1.1 Syväoppiminen	2
1.2 Piikkineuroverkko	3
1.2.1 STDP	4
1.2.2 Jännitepiikilliset neuronimallit	5
2 Neuromorfiset systeemit	9
2.1 Biologinen lähestymistapa	9
2.2 Ohjaamaton oppiminen	10
2.3 Hahmontunnistus	11
3 Neuromorfisten piirien fysikaalinen toteutus	13
3.1 Pii-fotoniikka	13
3.2 Piineuronit	13
3.3 Memristorit ja RSFQ	15
4 Nykyiset laajat toteutukset	17
4.1 SpiNNaker	17
4.2 IBM TrueNorth	18
5 Yhteenveto	20

Johdanto

Tieto- ja viestintäteknikka (ICT) on nykyään osa arkielämän jokaista puolta. Automaatio, koulutus, terveydenhuolto, kuljetus ja moni muu ala ovat yhteydessä ICT:n kehitykseen ja jopa riippuvaisia siitä. Mooren laki on mahdollistanut näiden teknologioiden pikaisen kehityksen, mutta valtavan energiatarpeen kustannuksella [1]. Yhteiskuntien digitalisaation seurauksena luodaan kiihtyvään tahtiin valtavia määriä dataa, joka pitää saada ihmisille hyödylliseen muotoon. Ratkaisuksi on luotu tämän datan käsittelyyn koneoppimisen ja tekoälyn algoritmeja [2].

Tekoälyalgoritmit ovat olleet tehokkaita keräämään hyödyllistä tietoa suurista määristä dataa käyttäen hyväksi neuroverkkoja (NN:ja) ja syväoppimista (DL:a). Niiden kouluttamiseen vaaditun energian määrä perinteisillä tietokoneilla ei ole kuitenkaan kestävää eikä ympäristöystävällistä, koska nykyiset laskennalliset menetelmät eivät sovellu hyvin NN:ien rinnakkaislaskentaan. Nykyisen tekoälyneuroverkon kuten GPT-3:n kouluttaminen tuottaa jopa yli 35 000 kg hiilidioksidipäästöjä. Biologiset aivot sen sijaan ovat huomattavasti tekoälysystemejä energiatehokkaampia ja oppivat helpommin varsinkin odottamattomissa ja uusissa tilanteissa. Jotkin tehtävät saattavat olla erittäin vaativia tekoälylle mutta vaivattomia aivoille. [1]

Mooren lain lähestyessä loppuaan yksi tapa kamppailla tätä laskennan energiaongelmaa on inspiroitua aivoista ja luoda uusi biologinen näkökulma laskentaan, jotta yhteiskuntien kehitys voi jatkuu.

1 Neuroverkot

Neuroverkot (NN:t) koostuvat toisiinsa kytköksissä olevista neuroneista, jotka ovat yksinkertaisia prosessoreita. Neuronit aktivoituvat eri syistä, sisääntuloneuronit ympäristöä havaitsevien antureiden avulla, ja muut neuronit aktivoituvat ympärillensä aktivoituvien neuronien kautta. Joidenkin neuroneiden aktivoituminen saattaa laukaista toimia. Neuroverkoille pyritään opettamaan synaptiset painot, joiden avulla ne toimivat halutulla tavalla ja tietävät milloin aktivoitua. Tämän avulla saadaan NN tekemään haluttu toimi esim. auton ajaminen. [2]

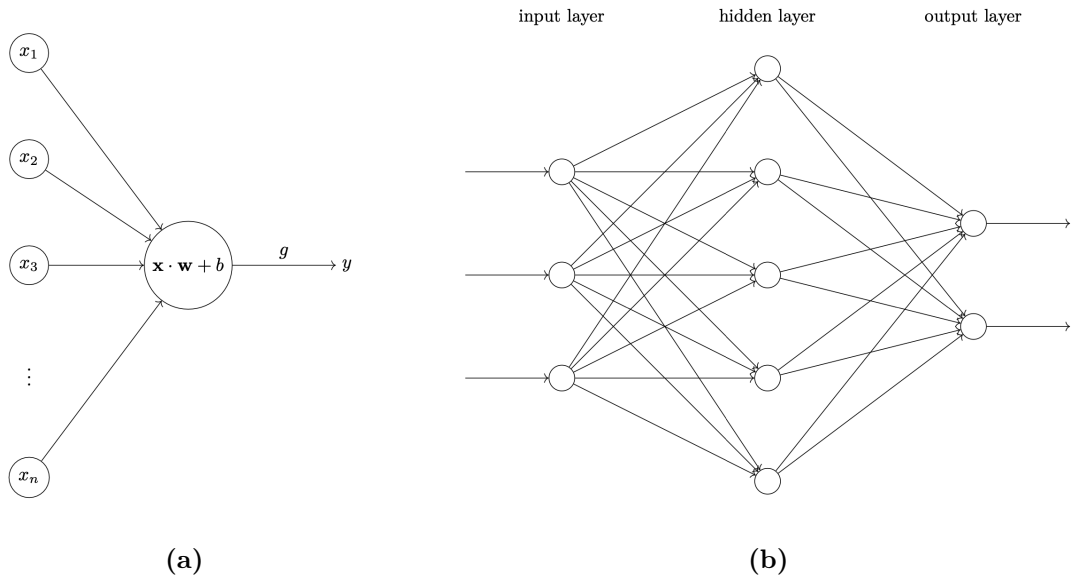
1.1 Syväoppiminen

Perseptroni on keinotekoinen neuroni ja NN yksinkertaisimmassa muodossaan, ja sitä pidetään syväoppimisen pohjana. Kuvassa 1a esitetty perseptroni vastaanottaa syötteen \mathbf{x} , ja muodostaa pistetulon painon \mathbf{w} kanssa, johon lisätään ennakkovakio b . Tämä lähetetään aktivaatiofunktiolle g , joka on sopiva tietylle sovellukselle sopiva muunnosfunktio. Perseptronin toiminta saa siis muodon

$$y = f(\mathbf{x}; \mathbf{w}, b) = g(\mathbf{x} \cdot \mathbf{w} + b), \quad (1)$$

ja tämä lopullinen tuloste saattaa toimia seuraavan neuronin syötteenä. [3]

Kytkenöille tai synapseille asetetut painot, jotka ovat reaaliarvoisia ja mahdollisesti muokattavia, määräävät kuinka se käyttäytyy ja oppii [2, 4]. Syvissä neuroverkoissa on useita kerroksia, joissa kaikki yhden kerroksen neuronit ovat yhteydessä seuraavan kerroksen neuroneihin kuvan 1b osoittamalla tavalla [5]. Syväoppiminen on oikeiden tarkkojen painojen asettamista syvälle neuroverkolle, jotta se tuottaisi haluttuja tuloksia [2].



Kuva 1: (a) Yksittäisen keinotekoisen neuronin eli perseptronin toimintamalli kaavan 1 osoittamalla tavalla. (b) Malli neuroverkon toiminnasta, jossa on vasemmalla syötekerros, keskellä piilokerros ja oikealla tulostekerros. [3]

Keinotekoisia neuroneita ja niille asetettuja painoja on siis DL:n toteuttavassa DNN:ssa monissa kerroksissa, ja termi syväoppiminen pohjautuukin näiden kerrosten ja kausaalisten ketjujen syvyydestä [2, 4]. Tämä DL:n hyödyntämä monikerroksinen arkkitehtuuri on tuottanut loistavia tuloksia koneoppimisen ja tekoälyn aloilla [4]. Laskelman syvyys ei kuitenkaan tee siitä suoraan vaikeaa, ja jotkin NN:t pystyvät nopeasti ratkaisemaan syvänkin laskelman esimerkiksi parametreja arvaamalla [2].

1.2 Piikkineuroverkko

Keinotekoiselle neuroverkolle (engl. *artificial neural network*, ANN) on tyypillistä yksittäinen jatkuva aktivaatio, joka toimii laskemalla arvon tietyllä syötteellä. Tämä on ristiriidassa neuromorfisten ja biologisten neuronien kanssa, jotka välittävät ja laskevat tietoa jännitepiikkien avulla. Neuronit eivät siis oikeasti keskustele toistensa kanssa jatkuvasti vaan lähettävät jännitepiikin kun sisäinen ajallinen dynamiikka

täyttää tietyt ehdot kuten kun kynnys ylittyy [2]. Tätä aivojen ominaisuutta imitoivaa neuroverkkoa kutsutaan piikkineuroverkoksi, (engl. *spiking neural network*, SNN) ja ne käyttävät näitä biologisesti inspiroituneita diskreettejä piikkejä tiedonkuljettajana. Nämä neuroverkot mahdollistavat siirtymisen pois analogisista neuroverkoista [6, 7], ja SNN:ja pidetään korkean suorituskyvyn oppimisjärjestelmänä [8].

Tarkemmin aivoissa olevat neuronit kommunikoivat piikkijonojen avulla, jotka ovat jono toimintapotentiaaleja. Piikeissä on paljon informaatiota, ja ne ovat ajallisesti harvassa, mikä tekee niistä neuroverkolle energiatehokkaan tavan kuljettaa tietoa. Perinteiset syvät neuroverkot vaativat korkean luokan näytönohjainkortteja, mutta SNN:ien avulla energiaa voidaan säästää. [4]

Oppiminen toteutuu kaikissa ANN:issa asettamalla synaptiset painot, joiden avulla neuroverkko osaa toimia. Piikkien käyttö SNN:ssa mahdollistaa kuitenkin biologisesti täsmällisemmät säännöt oppimiselle, ja näitä sääntöjä ja niiden muunnelmia kutsutaan hermopiikkien ajasta riippuvaksi plastisuudeksi (engl. *spike-time dependent plasticity*, STDP). [4]

1.2.1 STDP

STDP on ohjaamattoman oppimisen malli ja biologisesti vakuuttava mekanismi synaptiseen oppimiseen, mitä käytetään usein SNN:ssa [4, 6]. Malli hyödyntää aivojen synaptista plastisuutta, minkä avulla synapsit pystyvät heikentymään tai vahvistumaan ajan myötä [9]. Kuvassa 1b nähdään toisiinsa kytköksissä olevia neuroneita. Niistä signaalin lähettävä on presynaptinen neuroni ja signaalin vastaanottava postsynaptinen neuroni. STDP:ssa mikäli presynaptinen neuroni laukeaa tietyn ajan sisällä postsynaptisesta neuronin jälkeen, niiden välinen yhteys vahvistuu. Jos tietty postsynaptinen neuroni taas laukeaa ennen tiettyä presynaptista, niiden välinen yhteys heikentyy [4]. Näin verkon painot päivittyvät itsestään ja tämä ominaisuus

auttaa verkkoja kehittymään ajan myötä [9].

Kokeellisesti saatu yleisin muoto STDP:lle on

$$\Delta w = \begin{cases} Ae^{-\frac{-(t_{pre}-t_{post})}{\tau}} & t_{pre} - t_{post} \leq 0, A > 0 \\ Be^{-\frac{-(t_{pre}-t_{post})}{\tau}} & t_{pre} - t_{post} > 0, B < 0, \end{cases} \quad (2)$$

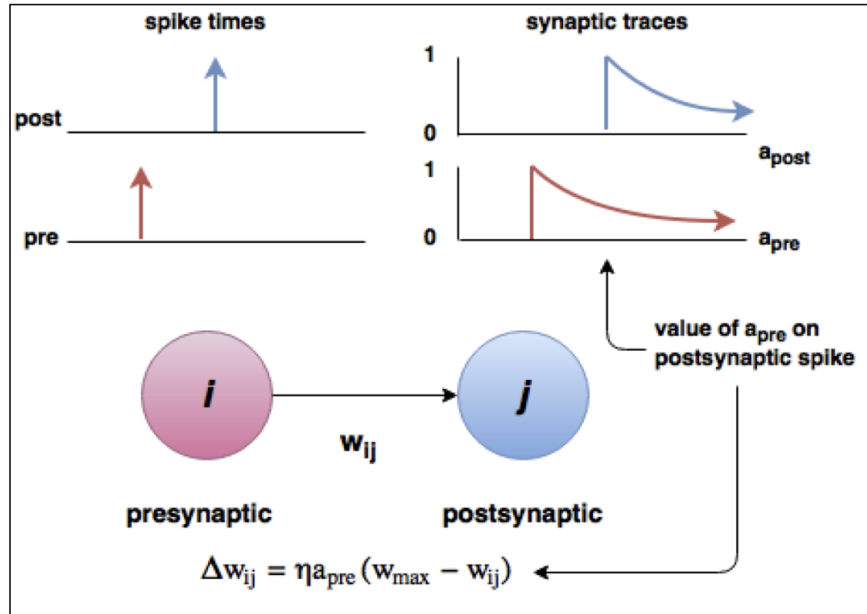
jossa w on synaptinen paino. A ja B ovat vakioita, joilla voidaan säätää oppimista. Suureet t_{pre} ja t_{post} osoittavat synapsien laukeamishetken aikana. τ on aikavakio, joka määrää kuinka nopeasti synaptisen muutoksen voimakkuus pienenee kun piikkien välinen aikaero kasvaa. Kaavan ylempi osuus käsittelee yhteyksien vahvistumista ja alempi osuus yhteyksien heikentymistä [4]. Tämä STDP-säännön painojen vahvistuminen ja heikentyminen laskee eksponentiaalisesti ajan funktiona, mikä nähdään kuvasta 2. SNN:t eivät usein käytä suoraan kaavaa (2) vaan muokkaavaat sen sopivaksi [4]. Kuva 2 esittää yhden mahdollisen kaavan.

STDP-sääntö keskittyy syötteen ensimmäiseen piikkiin, joka sisältää suurimman osan esimerkiksi hahmontunnistukseen tarvittavasta tiedosta. Toistuvien syötteiden esittäminen STDP:lla varustetulle SNN:lle muokkaa neuroneiden selektiivisyyttä SNN:n syötekuvioiden suhteen. Lisäksi STDP:n edetessä postsynaptinen neuronilaukeaa nopeammin tutulla syötteellä, mikä johtaa nopeampaan hermokäsittelyyn. Tämän avulla saadaan STDP:a hyödyntämällä neuronit reagoimaan tiettyyn syötteeseen nopeammin kuin muihin [4].

STDP:lla koulutetut SNN:t kärsivät kuitenkin ilmiöstä, jossa uuden tiedon oppimisen yhteydessä vanha tieto unohtuu. Tätä kutsutaan katastrofiseksi unohtamiseksi [6]. STDP on siis kaiken kaikkiaan loistava työkalu piikkineuroverkkojen oppimiseen, jonka ongelmapuolia täytyy silti vielä kehittää.

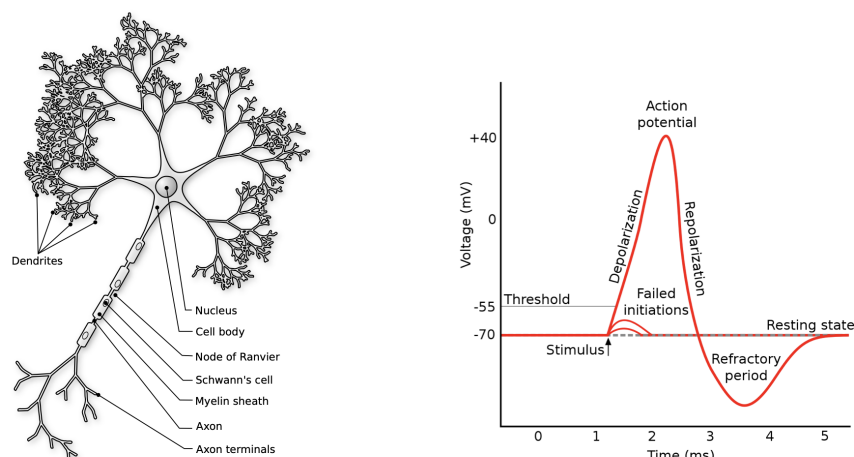
1.2.2 Jännitepiikilliset neuronimallit

Biologiset neuronit ovat soluja, jotka kommunikoivat keskenään sähkökemiallisten signaalien avulla. Näitä jännitepiikeistä koostuvia signaaleja luo natrium- ja ka-



Kuva 2: Mallikuva STDP-säännön vaikutuksesta painoihin. Synaptisten painojen etumerkkiin vaikuttaa synapsien laukeamisjärjestys. Painojen suuruuteen vaikuttaa syötteiden suuruus, joka laskee eksponentiaalisesti. [10]

liumionikanavat [11]. SNN:ssa olevien neuronien halutaan viestivän biologisten neuronien tavoin, joiden toimintaperiaate on esitetty kuvassa 3. Neuronin toiminnalle on esitetty erilaisia malleja, joista biologisesti lähimpänä on Hodgkin-Huxley (HH) malli [12]. Malli koostuu kuitenkin neljästä epälineaarista tavallisesta differentiaaliyhtälöstä, mikä tekee siitä hyvin tarkan, mutta laskennallisesti työlään. HH-malli ei tämän takia sovellu suurille verkoille [11]. Kuvassa 4 on HH-mallin kytkentäkaavio, jossa esimerkiksi solujen natrium- ja kaliumionien ionikanavat on esitetty konduktiivisuuksilla g_{Na} ja g_K [13]. Tämä ionikanavien toiminnan sisällytys mahdollistaa kuvassa 3 nähtävien jännitepiikkien tarkan mallinnuksen. Kuitenkin jos jokainen neuroni on näin monimutkainen, vaatii verkko valtavan määrän energiaa. On siis löydettävä tasapaino biologisen tarkkuuden ja energiavaatimusten kanssa, jotta neuromorfisen laskenta tärkeänä päämääränä oleva energiatehokkuus saavutetaan.



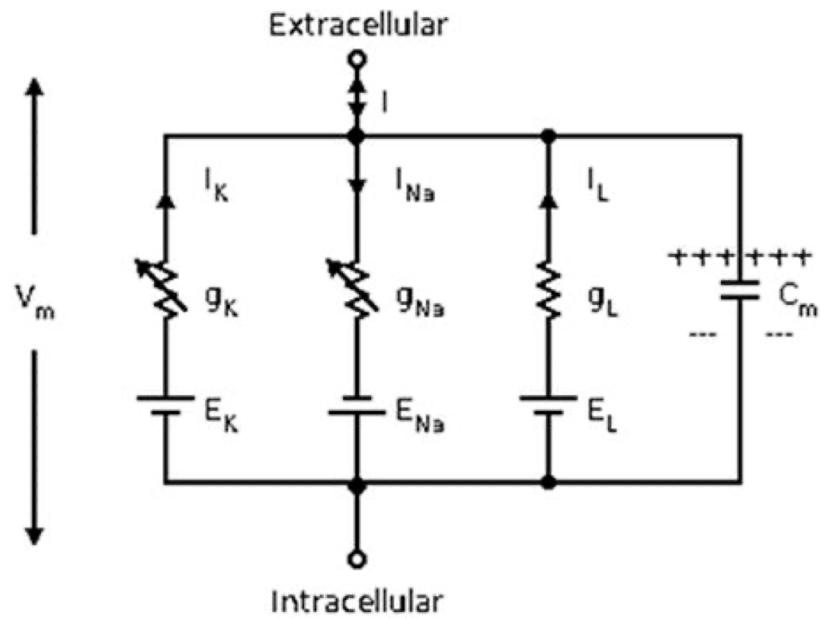
Kuva 3: Havaintokuva biologisesta neuronista ja sen lähettämästä jännitepiikistä.

Vasemmalla kuvassa biologinen neuroni. Sen dendriitit (dendrites) vastaanottavat signaaleja, jotka ovat saapuneet muilta neuroneilta, ja kuljettavat ne solun runkoon (cell body). Tarpeeksi vahva signaali luo siellä jännitepiikin, joka kulkeutuu aksonin (axon) kautta seuraavaan neuroniin. Oikealla kuvassa jännitepiikki, joka muodostuu ja laukeaa tarpeeksi vahvalla signaalilla kun tietty kynnyks ylitetty. Piikkiä seuraa hetki, jolloin jännite on lepotilaa alemmalla tasolla. [11]

Leaky-integrate-and-fire (LIF) mallissa systeemin neuronit pystyvät keräämään syötteitä kunnes tietty kynnyksarvo ylittyy, mikä nähdään kuvasta 5. Kynnyksarvon ylittyessä neuroni laukeaa ja antaa tulosteen, josta saattaa seurata ajanjakso, jolloin neuroni on lepotilassa eikä vastaanota hetkeen aikaan syötteitä [6]. Kuvasta nähdään myös laskevana käyränä, että mikäli LIF neuronin vastaanottama potentiaali ei saavuta kynnyksarvoa, se vuotaa hiljalleen, mikä vastaa biologisen neuronin toimintaa [12]. LIF-mallia voidaan kuvata differentiaaliyhtälöllä

$$C_m \frac{du}{dt} = g_l [E_l - u(t)] + I_{ext}(t) + I_{syn}(t), \quad (3)$$

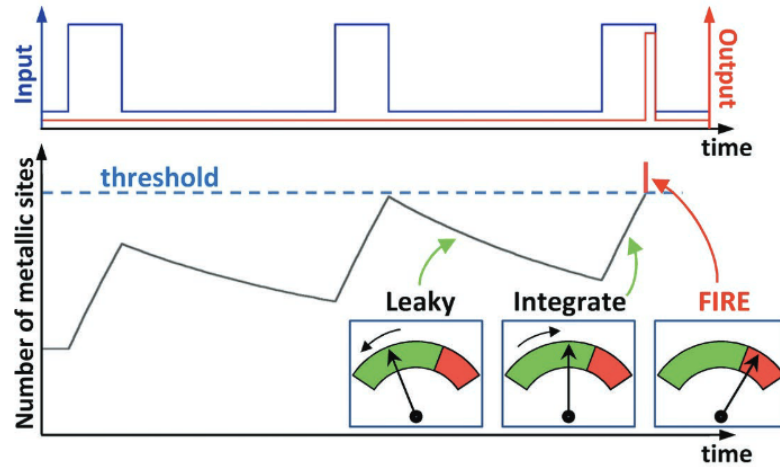
jossa u on solukalvon jännite, E_l on lepotilan jännite, C_m on solukalvon kapasitanssi ja I_{ext}, I_{syn} ovat syötteitä. Suuri ero HH-malliin on nyt kaavassa esiintyvä vuotokonduktiivisuus g_l , joka vetää jännitettä lepotilaa kohti ja korvaa HH-mallin



Kuva 4: Hodgkin-Huxley mallin kynkentäkaavio. [13]

monimutkaiset ionikanavat. [11]

LIF-malli esittää integroinnin ja laukaisun HH-mallia selkeämmällä ja helpommin esitettävällä tavalla hyödyntäen sitä tietoa, että jännitepiikkien muoto on yleensä sama ja suurin osa informaatiosta on piikin ajoituksessa [11, 12]. LIF-mallilla ei siis voida mallintaa jännitepiikkien muotoja vaan niitä pidetään abstrakteina tapahtumina. Näin laskentateho ei kuitenkaan kulu jännitepiikkien muodon selvittämiseen [11].



Kuva 5: Mallikuva LIF-neuronin toiminnasta ja syötteen vaikutuksesta tulosteeseen. Neuronin laukeaa ja lähettää jännitepiikin tulosteena kun tietty kynnyks ylitetty. Jos syötteen signaali ei ylitä kynnyks, se vähenee ajan mukana. Uusi signaali integroituu edelliseen jäljelle jääneeseen signaaliin. [14]

2 Neuromorfiset systeemit

Neurotieteen inspiroiman neuromorfisen laskennan tavoitteena on luoda keino-tekoiisiin hermosoluihin/synapseihin ja niiden vuorovaikutukseen perustuva uusi si-ruarkkitehtuuri, aivojen kaltaisen havainnoinnin ja oppimisen toteuttamiseksi [15]. Termin "neuromorfinen" loi Carver Mead 1980-luvun lopulla viittaamaan hermojär-jestelmiin, joiden arkkitehtuuri pohjautuu biologisiin hermostoihin [16].

2.1 Biologinen lähestymistapa

Vaikka ANN:t ovat jossain määrin aivojen inspiroimia, niitä ja erityisesti niiden kouluttamiseen käytettyä virheen takaisinetenemismekanismia (engl. *backpropagation mechanism*) on pidetty biologisesti epätodennäköisinä pitkään. Siitä huolimatta sekä laskennallisen neurotieteen että tekoälyn alalla on ollut suurta kiinnostusta löytää vaihtoehtoisia formulaatioita tai likiarvoja virheiden takaisinetenemiselle, mitkä käsittelevät alkuperäisen version biologisesti epätodennäköisiä puolia. [11]

DL on saatu aikaan rinnakkaislaskennan avulla, jossa yksi laskenta suoritetaan samanaikaisesti usealla suorittimella, minkä avulla se voidaan ratkaista nopeasti [6]. Rinnakkaislaskenta sisältyy von Neumann arkkitehtuuriin, jossa muistiyksikkö ja prosessointiyksikkö ovat erillään. Tämän kaltaisessa arkkitehtuurissa valtava määrä dataa on kuljettava yksiköiden välissä. Se vaatii paljon energiaa ja latenssista tulee myös ongelma [17]. Tämä eroaa suuresti aivojen energiaa säästävistä prosessointiominaisuuksista [6].

Neuromorfisen tietokoneen ohjelmointi vaatii yleensä siihen SNN:n luomista. SNN:t ovat inspiroituneet aivoista muita neuroverkkoja paremmin [18]. Merkittävänä erona muihin neuroverkkoihin on se, että SNN:ssa piikkien nopeuksien lisäksi myös piikkien ajoituksilla on merkitystä [2]. Tämä SNN:lla varustettu neuromorfinen systeemi mahdollistaa sopeutuvaa oppimista käyttäen vähän energiaa ja pinta-alaa. Tämän seurauksena ne ovat olleet lupaava vaihtoehto tulevaisuuden laskennalle [15].

2.2 Ohjaamaton oppiminen

Ohjaamattoman oppimisen toteutus keinotekoisien synaptisten ristikkorakenteiden avulla SNN:ssa on herättänyt viime aikoina suurta kiinnostusta [19]. Se käsittelee nimeämätöntä dataa, jolloin kone voi löytää tuntemattomia kuvioita ja ominaisuuksia itsenäisesti. Se havaitsee esimerkiksi klusterointia ja löytää yhteyksiä sekä poikkeamia datasta [15, 20]. STDP-sääntöä on hyödynnetty ristikkorakenteissa sijaitsevien muistialkioiden painojen päivityksessä ja ohjaamattoman oppimisen demonstroinnissa. Säännön avulla on pyritty jäljittelemään jo hyvin tunnettua biologisten neuroverkkojen oppimismallia [19].

Klusterointi onkin ohjaamattoman oppimisen pääprosessi, jossa samankaltaiset nimeämättömät alkiot jaetaan omiin alaluokkiinsa. Yksi yksinkertainen klusterointialgoritmi on kokoava hierarkkinen klusterointi (AHC). Jokaisessa vaiheessaan AHC luo alkiosta kaikki mahdolliset parit ja yhdistää yhdeksi klusteriksi parit, jotka muis-

tuttavat toisiaan eniten. [21]

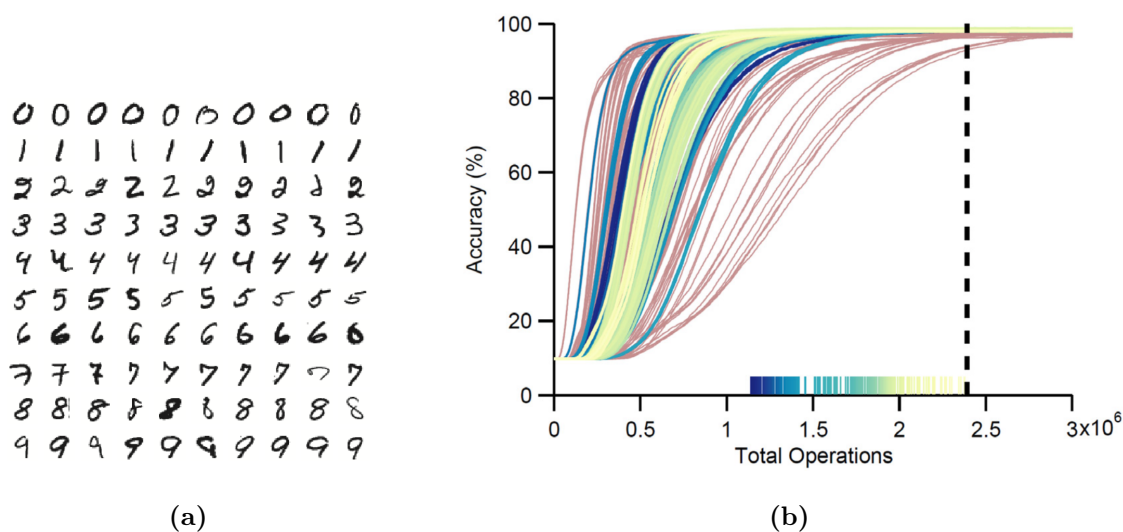
Vaihtoehtoinen ohjattu oppiminen yleensä koulutetaan tiettyyn tehtävään suurella merkityllä tietojoukolla, joka jaetaan satunnaisesti koulutusta, validointia ja testiä varten. Ohjattu oppiminen on kuitenkin kohtaamassa ongelmia, sillä se on vahvasti riippuvainen kalliista manuaalisista merkinnöistä, ja kärsii yleistysvirheistä sekä vääristä korrelaatioista. [20]

2.3 Hahmontunnistus

SNN:a on alettu huomioimaan lähiaikoina monilla hahmontunnistuksen aloilla kuten lääketieteellisissä diagnooseissa, puheentunnistuksessa sekä visuaalisessa prosessoinnissa. Aivojen alueissa, jotka käsittelevät aistitietoja sekä hermomotoristen ohjausalueiden ohjelmoinnissa piikkien ajallinen malli on jopa ratkaisevaa [4]. Monissa tapauksissa, kuten itsestään ajavan auton ohjelmoinnissa, on äärimmäisen tärkeää luotettava reaaliaikainen hahmontunnistus. Perinteisillä prosessoreilla saattaa olla vaikeuksia saavuttaa matala virrankulutus luotettavalla tavalla. Lisäksi neuro-morfiset systeemit kykenevät monimutkaiseen aivojen tapaiseen rinnakkaiseen laskentaan, mikä saattaa olla ratkaiseva tekijä monien hahmontunnistusta käyttävien järjestelmien kehityksessä [9].

Ohjaamattomalla oppimisella koulutettuja SNN:ja on toistaiseksi hyödynnetty matalatasoisten ominaisuuksien poiminnassa kuvista tunnistamista varten, ja tiedon ryhmittelyssä. Ne eivät kuitenkaan kykene vaativiin kognitiivisiin tehtäviin. [6]

Mixed National Institute of Standards and Technology (MNIST) tietokantaa on 1998 lähtien käytetty koneoppimisen ja hahmontunnistuksen testaamisen alustana. Tietokannan 70 000 esimerkistä 60 000 on kouluttamista ja 10 000 testaamista varten. Kirjoitusesimerkkejä on kerätty sekä lukio-opiskelijoilta että väestönlaskentatoimiston työntekijöiltä [22]. Kuvassa 6a on esitetty 100 esimerkkikuvaa, joita MNIST käyttää [22]. Kuvan 6b kuvaajasta nähdään, että SNN:it saavuttavat ha-



Kuva 6: Mixed National Institute of Standards and Technology (MNIST) tietokannan tuloksia koneoppimisen ja hamontunnistuksen testauksessa. (a) 100 esimerkkiä testaamisessa ja koulutuksessa käytetyistä kuvista [22]. (b) Kuvaaja operaatioiden määrästä, jotka tarvitaan saavuttamaan tietty tarkkuus MNIST-testissä [4]. Pystysuora katkoviiva näyttää kohdan, jossa verkot, jotka eivät käytä jännitepiikkejä saavuttava 98 % tarkkuuden. Muut käyrät näyttävät 522 syvän SNN:in tarkkuudet. Alla olevista pystysuorista värikkäistä viivoista nähdään kohdat, joissa SNN:it saavuttivat 98 % tarkkuuden. Vaaleanpunaiset käyrät eivät saavuttaneet 98 % tarkkuutta ennen verkkoja, jotka eivät käytä jännitepiikkejä.

montunnistuksessa 98 % tarkkuuden käyttäen vähemmän laskutoimituksia verrattuna perinteisiin neuroverkkoihin, jotka eivät käytä jännitepiikkejä [4]. Biologisesti inspiroituneiden neuroverkkojen tehokkuus hahmontunnistuksessa on ilmeinen.

3 Neuromorfisten piirien fysikaalinen toteutus

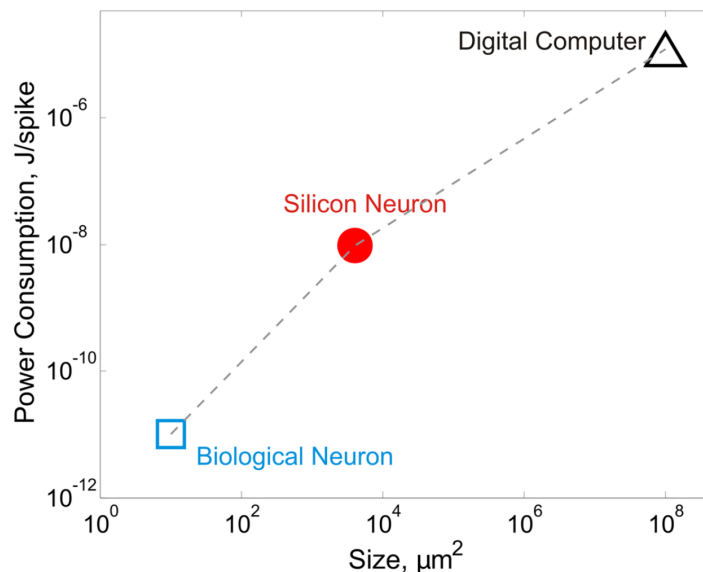
On vahvoja oletuksia, ettei pystytä enää nostamaan laskentatehoa pienentämällä komplementaaristen metallioksidipuoliyohteiden (CMOS:en) kokoa, joten toinen keino on löydettävä. Laskentatehon tarve on noussut 300 000 kertaisesti vuodesta 2012 vuoteen 2020 mennessä, ja tarpeen kasvu kiihtyy. Transistoreiden skaalaamisen sijaan voidaan myös tutkia uusia teknologioita. [17]

3.1 Pii-fotoniikka

On lukuisia keinoja toteuttaa neuroverkko tietokoneissa, mutta yhtenä lupaavimmista keinoista pidetään pii-fotoniikkaa, jossa yhdistetään elektroniset ja fotoniset piirit. Fotonisessa piirissä elektronien sijasta fotonit toimivat informaation kuljettajina, mikä on hyödyllistä, koska fotoneilla on enemmän hyödynnettäviä ulottuvuuksia. Valolla on esimerkiksi polarisaatio ja aallonpituus, joita käyttämällä voidaan keksiä uusia tapoja lähestyä neuromorfista laskentaa ja syväoppimista. Pii-fotoniikka ja sen komponentit ovat myös yhteensopivia nykyisten CMOS-piirien kanssa. Nämä yhdistetyt piirit sallivat laajan rinnakkaisen prosessoinnin, mikä saa aikaan nopeita ja virtaa säästäviä laskelmia. Tekoäly kehittyy jatkuvasti, ja täten DL vaatii enemmän suorituskykyä ja tehokkaampaa laskelmointia, johon edes pii-pohjaiset perinteiset komponentit eivät kykene. Pii-fotoniikkaa pidetään tällä hetkellä yhtenä lupaavimmista vaihtoehtoista neuromorfisen laskennan toteutuksessa. [23]

3.2 Piineuronit

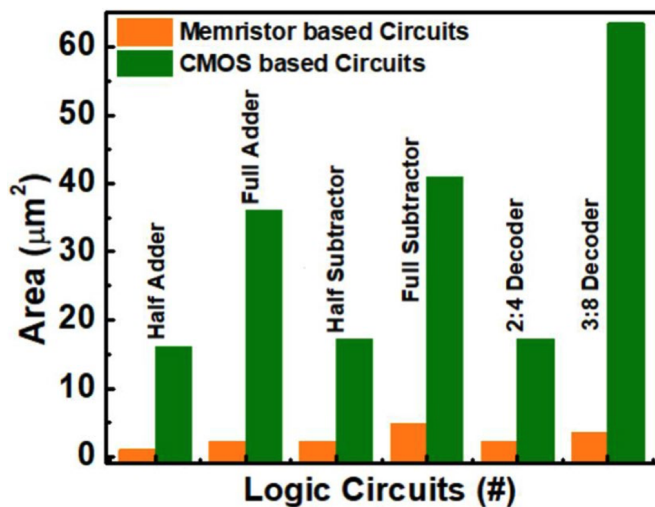
Piineuronit (SiN) ovat erittäin suuressa mittakaavassa integroituja piirejä (VLSI), jotka jäljittelevät biologisten neuronien sähköfysiologisia ominaisuuksia. Ne ovat sekä analogisia, että digitaalisia piirejä [16]. Piiri saattaa sisältää miljoonia tai jopa miljardeja CMOS transistoreita. Puolijohdepiitä käytetään aloitusmateriaalina suu-



Kuva 7: Biologisen neuronin (sininen neliö), pii-neuronin (punainen ympyrä) sekä digitaalisen tietokoneen (musta kolmio) virrankulutus nähden niiden viemään pinta-alaan [25].

rinnassa osassa integroituja piirejä, mutta puhdas pii ei johda hyvin sähköä, koska kaikki sen valenssielektronit ovat kemiallisissa sidoksissa. Johtavuutta parannetaan lisäämällä siihen epäpuhtauksia [24].

Neuromorfiset piirit, jotka käyttävät SiN:ta toimivat reaaliajassa, eikä verkon nopeus riipu hermosolujen lukumäärästä tai kytkennästä, mikä tekee niistä tehokkaita. Ne ovat paljon energiatehokkaampia ja tilaa säästävämpiä kuin yleiskäyttöiset tietokoneet [16]. Kuten kuvasta 7 nähdään, SiN:den ja digitaalisten koneiden viemien tilojen ero on jopa 4 kokoluokkaa [25]. Tämän takia SiN:it soveltuvat suuressa mittakaavassa toteutettaviin neuroverkkomulaatioihin [16]. Tätä neuromorfisten järjestelmien rakentamiseen luotua reaaliaikaista tekniikkaa voidaan hyödyntää käytännön sovelluksissa, joissa on hyötyä aivoja jäljittelevästä energiatehokkaasta laskennasta. SiN:sta rakennetut piirit ovat tästä syystä nykyään ratkaiseva osa neuromorfisten järjestelmien luonnissa ja niiden oppimisessa. [16]



Kuva 8: Eri logiikkapiirien kokoja [26]. Memristoreilla toteutetut piirit esitettynä oranssilla ja CMOS-piireillä toteutetut piirit esitettynä vihreällä. Piirit vasemmalta oikealle: puolisummain, summain, puolivähennin, vähennin, 2:4 dekooderi ja 3:8 dekooderi.

3.3 Memristorit ja RSFQ

Memristorit vaikuttavat yhdeltä hyvältä vaihtoehdolta korvata CMOS teknologia. Yksinkertaisimmillaan ja yleisimmässä muodossaan memristorit ovat vastuksia, joilla on muisti. Niiden resistanssi riippuu siitä millaisia sähköisiä signaaleja ne ovat saaneet. Ne ovat yksi tapa irtautua von Neumann arkkitehtuurista mahdollistamalla muistin sisäisen laskenta, jolloin ei jää tarvetta siirtää informaatiota prosessointiyksikölle ja takaisin. [17]

Memristorit ovat nanomittakaavassa sekä käyttävät hyvin vähän virtaa [27]. Pienien memristoripohjaisten ja CMOS-pohjaisten komponenttien kokoero nähdään kuvasta 8. Memristoreja pystytään myös ohjelmoimaan käytön aikana ja ne säilyttävät vakaasti resistanssin arvonsa [17, 27], mikä tekee niistä lupaavan vaihtoehdon neuromorfiselle laskennalle. Memristorien on myös huomattu imitoivan synapseja hyvin [28]. Memristiivisillä synapseilla on perinteiseen CMOS:een verrattuna mahdollisuus järjestää ne tiheään ristikkorakenteeseen, mutta nykyisin ne on silti integroituu

CMOS-piirien kanssa [27]. Memristoreita ja CMOS-piirejä voidaan näin käyttää yhdessä, jotta saadaan laaja integraatio aikaiseksi.

Suprjajohdetekniikalla on myös potentiaalia tuoda neuromorfiselle laskennalle parannuksia nopeuden ja energian säästön osa-alueilla. Suprajohtavien digitaalisten logiikkaperheiden osalta nopea yksittäisvuokvantti (engl. *rapid single flux*, RSFQ) logiikka tarjoaa todella suuren kytkentänopeuden ja alhaisen energiahäviön [29]. RSFQ:t käyttävät energiaa jopa 6 kertaluokkaa puolijohdetransistoreita vähemmän [30]. Suprajohteet vaativat kuitenkin matalia lämpötiloja, joten niitä voi olla hankala toteuttaa.

4 Nykyiset laajat toteutukset

Nykyisillä digitaalisilla neuromorfisilla systeemeillä on monia etuja analogisiin systeemeihin verrattuna, vaikka ne eivät olekaan yhtä energiatehokkaita. Digitaaliset systeemit pystyvät hyödyntämään yksinkertaisia yhtälöitä käyttämällä digitaalisia ohjelmia ja laitteistoja. Nykyisiä lupaavia digitaalisia systeemejä ovat esimerkiksi spiNNaker, IBM TrueNorth ja Intel Loihi. [31]

4.1 SpiNNaker

Piikkineuroverkkoarkkitehtuuri eng. spiking neural network architecture (SpiNNaker) projekti pyrkii luomaan massiivisesti rinnakkaisen miljoonaytimisen tietokoneen, joka on kytkentäarkkitehtuuriltaan neuromorfinen. Projekti soveltuu mallintamaan biologisia laajamittaisia SNN:ja [32]. Sen on suunnitellut Manchesterin yliopiston tietojenkäsittelytieteen laitoksen Advanced Processor Technologies Research Group (APT).

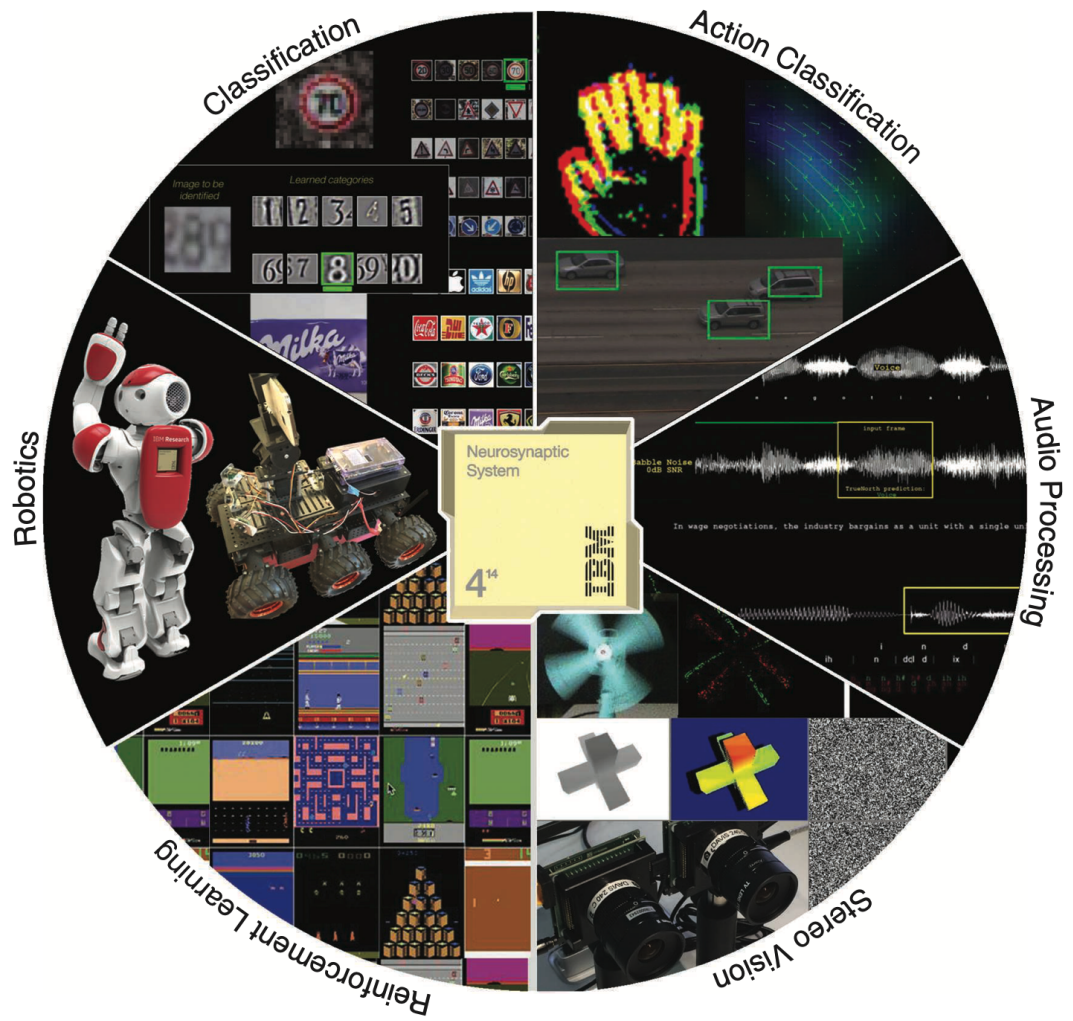
SpiNNakerin järjestelmän jokaisessa liitoksessa on kaksi piisirua. SpiNNaker siru sekä tietokoneissa käytettävä DDR-SDRAM-muistipiiri, jotka ovat yhdistetty johdimmilla [33]. Järjestelmä on suunniteltu ratkaisemaan pieniin itsenäisiin yksiköihin hajoitettuja tehtäviä, joilla ei ole keskusvalvojaa. Näitä tehtäviä kutsutaan yleisesti kiusallisen rinnakkaisiksi tehtäviksi (eng. *embarrassingly parallel problems*). Ohjelmisto antaa käyttäjän kuvata ohjelmaansa kaavioiden avulla. SpiNNakerin työkaluketjun on kirjoitettu pääasiassa Python ohjelmointikielen avulla. [32]

SpiNNaker on saanut merkittäviä asioita aikaiseksi aivojen jäljittelyssä. Sen järjestelmällä on mallinnettu esimerkiksi pikkuaivot. Lisäksi se on neuromorfinen järjestelmä, joka ensimmäisenä onnistui aivokuoren mallinnuksessa. [31]

4.2 IBM TrueNorth

TrueNorthia tutkitaan nykyään yli 50 laitoksessa yli 100 tutkijan toimesta. Sen sirussa on 4096 neurosynaptista ydintä. Jokainen ydin sisältää kaikki synaptiset painot, parametrit neuroneille sekä ohjeet piikkien toiminnalle. Ydin vastaanottaa 256 syötettä ja laskelmoi 256 tulostetta. Nämä ovat yhteydessä toisiinsa 256 x 256 ristikkorakenteella, jolloin littettävyys kaiken välillä säilyy ytimen sisällä. [34]

TrueNorthin sirujen avulla on rakennettu maailman suurin neurosynaptinen, eli aivojen toimintaa jäljittelevä, tietokone NS16e-4. Koko järjestelmässä on yhteensä 64 miljoonaa neuronua ja 16 miljardia synapsia neuronien välissä. Kone kykenee uskomattomaan $\sim 10^{11}$ synaptiseen toimeen sekunnissa/W, joten se soveltuu kuvasta 9 nähtäviin haastaviin ja monimutkaisiinkin käyttötarkoituksiin. TrueNorthin digitaalinen järjestelmä yhdistää von Neumann suorittimen neuromorfisen prosessorin kanssa, jolloin saadaan hyödyt molemmista järjestelmistä. [34]



Kuva 9: TrueNorth sirun käyttötarkoituksia. [34] Kuvassa esiintyy liikkeen luokittelu, audioprosessointi, stereonäkö, vahvistava oppiminen, robotiikka ja luokittelu.

5 Yhteenveto

Neuromorfiset systeemit ja niiden hyödyntämät piikkineuroverkot mahdollistavat erittäin energiatehokkaan laskennan vaihtoehdoksi perinteisille tietokoneille. Energian säästö ja komponenttien pieni mittakaava saattaa olla keino edistää tietotekniikkaa entisestään. Nämä ominaisuudet sallivat tehokkaan lisäksi myös huomattavasti ekologisemman laskennan. [15, 16, 27]

Neuromorfista laskentaa on lähiaikoina tutkittu paljon yksinkertaisilla tutkimuksilla, mutta on korkea aika yhdistää näistä saadut tiedot muihin osa-alueisiin. Koneoppiminen ja laskennallinen neurotiede on edistynyt huimaa tahtia. Nämä tiedot toisiinsa liittämällä on mahdollisuus ratkaista monet tämänhetkiset laskennalliset ongelmat, joita viestintä- ja informaatioteknologia alkaa kohdata. Tämä vaatii kuitenkin yhteistyötä monen erilaisen alan osajilta, mikä hidastaa edistymistä [1]. On äärimmäisen tärkeää, että yhteistyön ja sen kautta saatavien mahdollisten läpimurtojen tärkeys huomataan.

Tekoälyn käyttö tutkielmassa

Työssä on käytetty Google kääntäjää (Google translate) selkeyttämään joidenkin englanninkielisten lähteiden sisältöä. Tällä on pyritty varmistamaan, että asiat on ymmärretty oikein. Kääntäjällä käännettyä tekstiä ei ole kuitenkaan käytetty suoraan tutkielmassa, mutta sen antamat vastaukset ovat mahdollisesti vaikuttaneet kirjoitustyyliin ja sananvalintoihin. Yksittäisten sanojen kääntämiseen on pyritty käyttämään sanakirja.org ja termipankki.fi sivuja.

Työssä on käytetty myös OpenAI yhtiön GPT-3 ja GPT-4 malleja. Yksi käyttökohde on ollut oikolukeminen. Malleja on myös käytetty selventämään tutkielman sisällön aihealueita. Kysymykset ovat usein liittyneet suoraan lähteistä heränneisiin kysymyksiin, joihin ei ole löytynyt vastausta. Näillä tekoälyn tuottamilla vastauksilla on ainoastaan pyritty hahmottamaan aiheita, eikä tuotettujen vastausten sisältöä ole käytetty tutkielmassa.

Viitteet

- [1] G. Indiveri, *Neuromorphic Computing and Engineering* **1**, 1 (2021).
- [2] J. Schmidhuber, *Neural Networks* **61**, (2014).
- [3] J. Fassbender, Master's thesis, The university of Edinburgh, 2020.
- [4] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier ja A. Maida, *Neural Netw.* **111**, 47–63 (2019).
- [5] G. Indiveri ja S.-C. Liu, *Proceedings of the IEEE* **103**, 1379 (2015).
- [6] N. Rathi, I. Chakraborty, A. Kosta, A. Sengupta, A. Ankit, P. Panda ja K. Roy, *ACM Computing Surveys* **55**, (2022).
- [7] D. Auge, J. Hille, E. Mueller ja A. Knoll, *Neural Processing Letters* **53**, (2021).
- [8] T. Gao, B. deng, J. Wang ja G. Yi, *Frontiers in Neuroscience* **16**, 929644 (2022).
- [9] E. Elitalib ja A. Bahar, *Algorithm Asynchronous* **1**, 24 (2023).
- [10] D. Saunders, H. Siegelmann, R. Kozma ja M. Ruzinkao, kirjassa *STDP Learning of Image Patches with Convolutional Spiking Neural Networks* (PUBLISHER, 2018), pp. 1–7.
- [11] L. Kriener, väitöskirja, Graduate School for Cellular and Biomedical Sciences University of Bern, 2023.
- [12] H.-M. Huang, Y. Rui, Z. Tan, H. He, W. Zhou, J. Xiong ja X. Guo, *Advanced Materials* **31**, 1803849 (2018).
- [13] S. Zhu, *Journal of Physics: Conference Series* **2634**, 012044 (2023).
- [14] P. Stoliar, J. Tranchant, C. Benoit, E. Janod, M.-P. Besland, F. Tesler, M. Rozenberg ja L. Cario, *Advanced Functional Materials* **27**, 1604740 (2017).

- [15] X. Wan, W. Xu, M. Zhang, N. He, X. Lian, H. Ertao, J. Xu ja Y. Tong, *ACS Applied Electronic Materials* **2**, 3497 (2020).
- [16] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. D. Häfliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. V. Arthur, K. M. Hynna, F. O. Folorosele, S. Saighi, T. Serrano-Gotarredona, J. H. B. Wijekoon, Y. Wang ja K. A. Boahen, *Frontiers in Neuroscience* **5**, (2011).
- [17] A. Mehonic, A. Sebastian, B. Rajendran, O. Simeone, E. Vasilaki ja A. Kenyon, *Advanced Intelligent Systems* **2**, (2020).
- [18] C. Schuman, S. Kulkarni, M. Parsa, J. Mitchell, P. Date ja B. Kay, *Nature Computational Science* **2**, 10 (2022).
- [19] G. Malavena, M. Filippi, A. Spinelli ja C. Compagnoni, *IEEE Transactions on Electron Devices* **PP**, 1 (2019).
- [20] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang ja J. Tang, *IEEE Transactions on Knowledge and Data Engineering* **PP**, 1 (2021).
- [21] T. Jo, *Machine Learning Foundations: Supervised, Unsupervised, and Advanced Learning* (Springer Nature Switzerland AG, 2021).
- [22] A. Baldominos, Y. Sáez ja P. Isasi, *Applied Sciences* **2019**, 3169 (2019).
- [23] B. Xu, Y. Huang, Y.-T. Fang, Z. Wang, S. Yu ja R. Xu, *Photonics* **9**, 698 (2022).
- [24] D. M. H. Neil H. E. Weste, *CMOS VLSI Design: A Circuits and Systems Perspective, Fourth Edition* (Addison-Wesley, 2010).
- [25] C.-S. Poon ja K. Zhou, *Frontiers in Neuroscience* **5**, 108 (2011).
- [26] S. Ghodke, S. Kumar, S. Yadav, N. Dhakad ja S. Mukherjee, *Journal of Computational Electronics* **23**, 1 (2023).

- [27] M. Rahimi Azghadi, B. Linares-Barranco, D. Abbott ja P. Leong, IEEE Transactions on Biomedical Circuits and Systems **11**, 434–445 (2017).
- [28] T. Hynnä, Master’s thesis, University of Turku, 2022.
- [29] M. Jardine ja C. Fourie, IEEE Transactions on Applied Superconductivity **PP**, 1 (2023).
- [30] J. Huang, R. Fu, X. Ye ja D. Fan, CCF Transactions on High Performance Computing **4**, (2022).
- [31] S. Furber, National Science Review **11**, nwad283 (2023).
- [32] S. Furber, F. Galluppi, S. Temple ja L. Plana, Proceedings of the IEEE **102**, 652 (2014).
- [33] S. Furber, D. Lester, L. Plana, J. Garside, E. Painkras, S. Temple ja A. Brown, Computers, IEEE Transactions on **62**, 2454 (2013).
- [34] M. DeBole, B. Taba, A. Amir, F. Akopyan, A. Andreopoulos, W. Risk, J. Kusnitz, C. Otero, T. Nayak, R. Appuswamy, P. Carlson, A. Cassidy, P. Datta, S. Esser, G. Garreau, K. Holland, S. Lekuch, M. Mastro, J. McKinstry ja D. Modha, Computer **52**, 20 (2019).