



IKER
GAZTE
NAZIOARTEKO
IKERKETA EUSKARAZ

VI. IKERGAZTE

NAZIOARTEKO IKERKETA EUSKARAZ

2025eko maiatzaren 28, 29 eta 30a
Bilbo, Euskal Herria

ANTOLATZAILEA:
Udako Euskal Unibertsitatea (UEU)



Aitortu-PartekatuBerdin 4.0

INGENIARITZA ETA ARKITEKTURA

**Pultsito: Pultsu bidezko Neurona-
sareak erabiltzeko liburutegia C
programazio-lengoaian**

*Alex Amenabar, Unai Garciarena,
Javier Navaridas
eta Jose Antonio Pascual*

141-148 or.
<https://dx.doi.org/10.26876/ikergazte.vi.03.17>

ANTOLATZAILEA



BABESLEAK



EUSKO JAURLARITZA
GOBIERNO VASCO



LAGUNTZAILEAK



Pultsito: Pultsu bidezko Neurona-sareak erabiltzeko liburutegia C programazio-lengoaian

Alex Amenabar¹, Unai Garciaarena¹, Javier Navaridas¹, Jose A. Pascual¹

Intelligent Systems Group (ISG), Konputagailuen Arkitektura eta Teknologia Saila,
Euskal Herriko Unibertsitatea (UPV/EHU).
Manuel Lardizabal Pasealekua 1, 20018 Donostia.

alex.amenabar@ehu.eus

Laburpena

Pultsu bidezko Neurona-sareen (PNS) arloko ikerkuntzak hazkunde handia bizi izan du azken urteetan, sare horiek neurona-sare biologikoekin dituzten antzekotasunak direla eta. Hala ere, oraindik ez da lortu PNSekin sare klasikoekin lortu diren emaitzak berdintzea. PNSak biologikoki errealistagoak dira, baina entrenamendu prozesua eta informazioaren kodeketa konplexuagoak dira. Sare horiekin lan egiteko software tresnak ezinbestekoak dira, baina gehienak garapenean daude edo oso erabilera zehatzetara mugatzen dira. Lan honetan, ikaskuntza automatikoan eta simulazioan erabilgarriak diren PNSak sortu eta erabiltzeko liburutegi orokorrako baten oinarria garatu da. Inplementazioa C programazio-lengoaian egin da ahalik eta errendimendurik altuena lortzeko.

Hitz gakoak: Pultsoen Neurona-Sareak, Neurona-Sareak, Neurozientzia, C, Kode paralelismoa, OpenMP

Abstract

Research in the field of spiking neural network (SNN) has experienced a great growth in recent years, due to the similarities these networks have with biological neural networks. However, SNNs are yet to match the results obtained with the classics networks. While SNNs are more biologically realistic, the training process, information encoding, etc. are more complex. Software tools for working with these networks are essential, but most are under development or limited to very specific use cases. In this work, the basis of a more general library for creating and using SNN useful in machine learning and simulation has been developed. The implementation is done in the C programming language to achieve the best possible performance.

Keywords: Spiking Neural Networks, Neural Networks, Neuroscience, C, Code parallelism, OpenMP

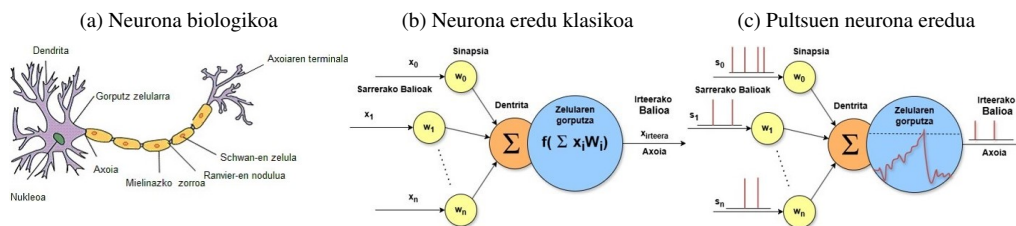
1 Sarrera eta motibazioa

Azken hamarkadan ikaskuntza automatikoak izugarrizko aldaketak eta aurrerapenak jasan ditu neurona-sare sakonen garapena dela eta. Eredu horiek, neurona-sare biologikoetan oinarrituta, duela urte gutxi arte pentsa ezinak ziren emaitzak lortu dituzte ataza desberdinetan, hala nola, lengoaia naturalaren prozesamenduan (Otter *et al.*, 2021), irudien prozesamenduan (Archana eta Jeevaraj, 2024), eta musika sortzeko tresnetan (Parker *et al.*, 2024). Halere, sare horien entrenamendu prozesua luzea da eta grafikoak prozesatzeko unitateen (GPU) erabilera intentsiboa ezinbestekoa da, energia kontsumoa izugarria izanik (Xu *et al.*, 2023). Bestalde, PNSek erantzunak neurona-sare sakonek baino azkarrago sortu ditzakete neurona-sare biologikoekin dituzten antzekotasunak direla eta. Gizakien garuna, denbora laburrean ataza konplexuak gauzatzeko gai da, energia gutxi erabiliz (Cox eta Dean, 2014). Neurona-sare sakonetan gertatzen ez den bezala.

Pultsu bidezko Neurona-sareak (PNS) neurona-sare biologikoetan oinarritzen dira hauen gaitasunak aprobe-txatzeko. Neurona-sare sakonak neurona-sare biologikoen egiturari oinarritzen dira, baina haien funtzionamendua guztiz desberdina da. PNSen funtzionamendua neurona-sare biologikoenetan oinarritzen da, informazioa pultsuen bidez prozesatuz.

Azken urteetan PNSen inguruko ikerkuntzak hazkunde handia izan du, baina oraindik ez dira neurona-sare sakonen emaitzak berdindu. Horren arrazoi nagusia sareen entrenamendu prozesuaren konplexutasuna da. Beste

1. Irudia: (a) irudian neurona biologiko bat ikus daiteke. (b) irudikoa, berriz, neurona-sare sakonetan erabiltzen den neurona eredua da. Sarrera, x_i , zenbakiak dira. Neuronak sarrera guztiak konbinatzen ditu eta funtzio baten bidez irteera kalkulatu. Azkenik, (c) irudian pultsuetan oinarritutako neurona eredu bat ikus daiteke. Kasu honetan, informazioa pultsuen bidez transmititzen da. Neuronak pultsuak jasotzean, bere mintz potentziala aldatzen da, atalasea gainditu eta pultsu bat sortu arte.



arrazoi bat software tresna eta gailu fisiko espezializatuen falta da. Neurona-sare klasikoaren entrenamendua azkartzeko gailu ugari garatu dira, baina PNSen kasuan garapena xumeagoa da.

Lan honetan, software tresnetan jarri da arreta, izan ere, nahiz eta tresna ugari dauden PNSekin lan egiteko, gehienak erabilera konkretuetara mugatuta edo garapenean daude. Gainera, software tresna asko Python programazio-lengoaian daude garatuta, errendimenduan eragin negatiboa duena, bereziki simulazio handiak gauzatzean. Hori dela eta, lan honetan liburutegi bat garatu da, helburutzat erabilera orokorra izatea duena. Liburutegia C programazio-lengoaian garatu da ahalik eta errendimendu altuena lortzeko.

2 Arloko egoera eta ikerketaren helburuak

2.1 Pultsu bidezko neurona-sareak

PNSetan, neuronek sare biologikoetan sortzen diren pultsuak simulatzen dituzte, pultsuak neuronaren irteerako lotura sinaptikoetatik (neuronen arteko loturak) bidaliz. Neurona bakoitzak sarrerako eta irteerako lotura sinaptikoak ditu, non sarrerako lotura sinaptikoetatik neuronak beste neurona batzuk sortutako pultsuak jasotzen dituen, eta irteerakoetatik, berriz, bidali. Pultsuek neuronaren mintz-potentzian eragiten dute. Mintz-potentziala neuronaren kanpo eta barnealdearen arteko boltaia diferentzia da. Neurona batek pulsu elektrikoak jasotzen dituenean boltaia desberdintasun hori areagotu egiten da, mintz potentzialaren balioa handituz. Balio hori atalase batera heltzen denean, neuronak pulsu elektriko bat bidaltzen du eta deskantsu fasean sartzen da denbora-tarte batez, non ez dituen sarrerako pulsuak prozesatzen eta mintz potentziala balio jakin batera murrizten den.

1. Irudian, (a) neurona biologiko baten, (b) neurona-sare sakonetako neurona ereduaren, eta (c) pultsuen neurona-sareetako neurona baten arteko desberdintasunak eta egitura ikus daitezke.

2.1.1 Neurona ereduak

PNSetan neurona eredu desberdinak erabiltzen dira, bi ezaugarriaren arabera neurtzen direnak: sinesgarritasun biologikoa eta kostu konputazionala. PNSak simulazioetarako eta ikaskuntza automatikorako erabiltzen dira, bi kasuetan helburuak desberdinak izanik. Simulazioetan, oinarri biologikoei garrantzi handia dute, eta beraz, horiek kostu konputazionalak baino garrantzi handiagoa du. Bestalde, ikaskuntza automatikoan ez da ezinbestekoa prozesu biologikoak guztiz modu zehatzean simulatzea, ondorioz, konputazionalki eredu sinpleagoak erabiliz.

Atal honetan neurona eredu bat aztertuko da: Leaky-Integrate-and-Fire (LIF) (Hunsberger eta Eliasmith, 2015). Eredu hau ikaskuntza automatikoan erabiltzen da kostu konputazional eta sinesgarritasun biologikoaren arteko oreka dela eta.

2.1.1.1 Leaky-Integrate and Fire: LIF neurona ereduak 1. eta 2. ekuazioen bidez deskribatzen da.

$$\tau \frac{dV}{dt} = V_{rest} - V + RI(t) \quad (1)$$

$$V = V_{rest} \text{ baldin } V \geq V_{th} \quad (2)$$

1. ekuazioan, τ balio konstanteak mintz potentzialeko aldaketak kontrolatzen ditu. Horren arabera, aldaketak handiagoak edo txikiagoak dira. V_{rest} bidez neuronaren mintz potentziala murrizten da denboran zehar. V uneko mintz potentziala da, R neuronaren erresistentzia (sarreraren intentsitatea kontrolatzen duena) eta $I(t)$, t unean neuronak jasotako pultsuen eragina. 2. ekuazioak neuronaren mintz potentziala atalasea heltzean gertatzen dena modelatzen du, neurona deskantsu fasean sartuz eta bere mintz potentziala V_{rest} baliora murriztuz. Atalasearen balioa V_{th} da.

2.1.2 Pultsu bidezko neurona-sareen ikaskuntza

Neurona-sare sakonetan ikaskuntza prozesua gradientearen jaitsieraren bidez gauzatu ohi da, baina PNSetan ezin da metodo hori zuzenean erabili, neuronon aktibazio funtzioa diskretua izanik (pultsuak), ezin baitira gradienteak kalkulatu. Gainera, gradiente jaitsiera ez da prozesu biologikoetan oinarritzen. Hala ere, teknikak neurona-sare sakonetan izandako arrakasta dela eta, PNSetara moldatzeko algoritmo asko inplementatu dira, adibidez (Perez-Nieves eta Goodman, 2021).

Gradientearen jaitsieraz gain, PNSen oinarri biologiko sendoak direla eta, biologian oinarritutako ikaskuntza prozesuak ere zehaztu dira. Hauetan, ezagunena Hebb-en teorian oinarritutako plastikotasun sinaptikoa da (Gerstner eta Kistler, 2002). Teoriaren oinarria, batera aktibatzen diren neuronon loturak indartu egiten direla da. Erregela horretatik epe luzerako indartzea (LTP ingeleseko sigletatik) eta epe luzerako ahultzea (LTD ingeleseko sigletatik) eratortzen dira. Lehenengoaren arabera, sinapsi osteko neuronak sinapsi aurrekoaren ostean sortzen badu pultsua, bien arteko lotura indartu egiten da. LTDren arabera, sinapsi osteko neuronak sinapsi aurrekoak baino lehenago sortzen badu pultsua, lotura ahuldu egiten da.

LTP eta LTDtik, pultsuaren unearen menpeko plastikotasuna deritzon ikaskuntza erregela eratortzen da (STDP ingeleseko siglengatik), LTP eta LTDk definitutako erregela modelatzen duena, baina pultsuen artean pasatako denboraren arabera indartze edo ahultzearen eragina kontrolatuz. Erregela horren formulazio orokor bat 3. ekuazioan islatzen da.

$$\Delta w = \begin{cases} A_+ \exp(-\frac{\Delta t}{\tau_+}) & \Delta t > 0 \\ -A_- \exp(\frac{\Delta t}{\tau_-}) & \Delta t < 0, \end{cases} \quad (3)$$

non A_+ , A_- , τ_+ eta τ_- pisuen aldaketak moldatzeko konstanteak diren eta Δt sinapsi osteko eta aurreko neuronek pultsua sortutako uneen arteko aldea.

2.1.3 Sare topologiak

Neurona-sarearen topologia neuronek eta hauen arteko loturek osatzen duten egitura da. Neurona-sare sakonetan topologiak nahiko egituratuak dira, normalean neuronak geruzetan antolatuta egonik eta neuronak hurrengo geruzetako neuronekin konektatuta. Hau ez da horrela neurona-sare biologikoetan, topologiak ez dira horren egituratuak. Ondorioz, PNSetan ere topologia aske edo arbitrarioak aurki daitezke.

2.2 Tresnak

PNSekin lan egiteko software tresna gehienak ikaskuntza automatikora edo simulazio biologikoak gauzatzera bideratuta daude, baina normalean ez dituzte ezaugarriak inplementatzen dituzte, baina ez dute aukera ematen biak gauzatzeko. Adibidez, ikaskuntza automatikora bideratutako tresnek neurona eredu sinpleak inplementatzen dituzte eta sare topologia askeak edo arbitrarioak erabiltzeko aukera eman gabe.

1. Taulan software ezagun batzuk eta haien ezaugarri garrantzitsuenak biltzen dira. Tresna horietan exekuzio mota erlojuaren arabera edo gertaeretan oinarritua izan daiteke. Erlojuan oinarrituta daudenen kasuan, simulazioa iterazioetan antolatzen da, non iterazio bakoitza denboran pauso bat den eta iterazio bakoitzean neurona guztiak prozesatzen diren. Gertaeretan oinarritutako inplementazioen kasuan, neuronak horiekin erlazionatutako gertaerak daudenean bakarrik prozesatzen dira.

Bestalde, 1. Taulan aurkeztutako teknologia gehienak erabilera zehatzetara mugatuta edo garapenean daude. Gainera, kasu batzuetan oinarrian erabilitako programazio-lengoaiak eragin negatiboa du errendimenduan. Hala ere, beharren arabera software egokiak dira.

1. Taula: Pultsuen neurona-sareak erabiltzeko liburutegi ezagunenak eta haien ezaugarri garrantzitsuenak.

Liburutegia / tresna	Exekuzio-motak	Neurona ereduak	Ikaskuntza erregelak	Programazio-lengoaia	Azelerazio paraleloa	Helburua(k)
snnTorch	Erlojuan oinarritua	LIF	Gradientearen jaitziera	Python	CUDA	IA
SpikingJelly	Biak	IF	Gradientearen jaitziera, STDP, ANN bihurketa	Python	CUDA	IA
Norse	Erlojuan oinarritua	Izkihevich, LIF, ...	STDP, gradientearen jaitziera	Python	CUDA	IA
NengoDL	Erlojuan oinarritua	LIF eta antzekoak	ANN bihurketa	C++	CUDA	
BindsNet	Erlojuan oinarritua	Izhikevich, LIF	Hebbian, STDP, R-STDP, R-STDP-et	C++, Python-eko adaptagailua	OpenMP, CUDA	Simulazioak
Brian2	Erlojuan oinarritua	Definitzekoa	STDP-et	C++	OpenMP	Simulazioak
NEST	Biak	LIF, HH, SRM, ...	STDP-et	C++, Python	MPI	Simulazioak
Pulstisito	Erlojuan oinarritua	LIF	STDP, Anti-STDP	C	OpenMP	IA, Simulazioak

2.3 Ikerkuntzaren helburuak

Lan honetan, PNSekin lan egiteko liburutegi baten lehen bertsioa garatu da, non sare arbitrarioak sortu daitezkeen, sare berean ikaskuntza-erregela desberdinak konbinatuz. Arreta berezia liburutegiaren errendimenduan jarrita, C programazio-lengoaia erabili da exekuzio denborak ahalik eta gehien murrizteko (ikus 4. atala). Gainera, simulazioen algoritmoa paralelizagarria da PUZean eta GPUan.

Liburutegi honen garapena bultzatu duen helburuetako bat beste lan batean aurkitzen da. PNSetan topologiak garrantzi handia izan dezake loturen askatasuna dela eta. Halere, ikaskuntza automatikoko atazetan normalean oso sare egituratuak erabiltzen dira, neurona-sare sakonetan egiten den antzera. Topologia kaotikoak eskuz sortzeko zailtasuna dela eta, PNSentzako arkitektura egokien bilaketarako algoritmoen arloko ikerkuntzak (NAS ingeleseko siglengatik) bultzada handia izan du azken urteetan (Shen *et al.*, 2024). Arlo horretan, hainbat lan burutu dira sare biologikoetan oinarritutako topologiak bilatzeko, adibide bat Pan *et al.* (2023) izanik, non sarearen oinarritzko elementu gisa neuronak erabili beharrean neuronez osatutako eta neurona-sare biologikoetan aurkitzen diren neuronen zirkuitu txikiak erabiltzen diren (Mittal eta Narayanan, 2024). Lan honetan garatutako liburutegia topologiaren bilaketarako lana gauzatzeko erabiliko da.

3 Ikerketaren muina

3.1 Inplementatutako ezaugarriak

Liburutegiaren lehenengo bertsioa izanik, inplementatutako ezaugarri gehienak oinarritzko kontuak dira. Exekuzio-motari dagokionez, erlojuan oinarritutako exekuzioa dago inplementatuta. Neurona ereduaren inguruan, LIF ereduak aurki daiteke liburutegian, baina datu-egiturek aukera ematen dute neurona eredu berriak kode orokorra aldatu gabe txertatzeko. Horretaz gain, neuronak kitzikatzaileak edo inhibitzaileak izan daitezke.

Ikaskuntza-erregelari dagokionez, STDP erregelaren bertsio desberdinak inplementatu dira. Bestalde, lotura sinaptiko bakoitzak bere ikaskuntza-erregela propioa izan dezake sare berean. Horretaz gain, lotura sinaptikoek latentzia dute (pultsuak sinapsi aurreko neuronatik sinapsi osteko neuronara iristeko behar duen denbora) eta bi neuronaren artean lotura ugari egon daitezke.

3.2 Datu-egiturak

Neurona-sareak kudeatzeko egitura bi zerrendatan oinarritzen da, neuronena, eta lotura sinaptikoena. Aldi berean, neurona bakoitzak sarrerako eta irteerako lotura sinaptikoen zerrendak eta beste hainbat datu ditu, hala nola, mintz potentziala, deskantsu fasean dagoen eta azken pultsua sortu duen unea.

Lotura sinaptikoei dagokionez, lotura bakoitzak pisua, latentzia, lotura aurreko neurona, lotura ondorengo

neurona eta prozesatu beharreko pultsuen zerrenda ditu. Lotura aurreko neuronak pultsu bat sortzen duenean, pultsua prozesatu behar den unea loturaren pultsuen zerrendan kokatzen da. Pultsua prozesatzeko unea, iterazioari dagokion unearen eta loturaren latentziaren arteko batura da.

3.3 Exekuzioaren algoritmoa

Inplementazioa erlojuan oinarritua denez, iterazio bakoitzean neurona guztiak prozesatzen dira. Neurona prozesatzean, hasteko, honen sarrerako lotura sinaptikoak aztertzen dira mintz potentziala eguneratzeko, eta, ondoren, pultsua sortzen bada, irteerako loturen zerrendetara gehitu pultsua. Azkenik, lotura guztiak prozesatzen dira ikaskuntza-erregelaren arabera pisuak eguneratzeko. 1. Algoritmoan deskribatutako prozesua islatzen da.

Algoritmoa 1 Denbora-pauso bakoitzeko prozesamendua erlojuan oinarritutako exekuzioan.

```

Bitartean t < denbora_pausoak egin
  For neurona: neurona zerrendan egin
    Prozesatu_sarrerako_loturak(neurona, t)
  Amaiera For
  For neurona: neurona zerrendan egin
    Prozesatu_irteerako_loturak(neurona, t)
  Amaiera For
  For lotura: loturen zerrendan egin
    Prozesatu_ikaskuntza_erregela(lotura)
  Amaiera For
Amaiera Bitartean

```

1. Algoritmoan, neuronaren prozesamendua bi ataletan banatzen da: sarrerako loturetatik mintz potentziala eguneratu, eta ondoren, aztertu pultsua sortu behar den edo ez. Bi prozesu horiek 2. eta 3. Algoritmoetan ikus daitezke. Bi funtzioek prozesatu beharreko neurona eta simulazioko denbora pausoa jasotzen dituzte parametro bezala.

Algoritmoa 2 *Prozesatu_sarrerako_loturak* funtzioaren algoritmoa.

```

Eskatu neurona, t
  bolt = 0
  For lotura_sinaptikoa: neuronaren sarrerako lotura sinaptikoak egin
    Baldin lotura_sinaptikoa → hurrengo_pultsoaren_unea == t orduan
      bolt += lotura_sinaptikoa → w
      eguneratu_loturaren_pultsoen_zerrenda(lotura_sinaptikoa)
    Amaiera Baldin
  Amaiera For
  eguneratu_neuronaren_mintz_potentziala(neurona)

```

Algoritmoa 3 *Prozesatu_irteerako_loturak* funtzioaren algoritmoa.

```

Eskatu neurona, t
  Baldin neurona → mintz_potentziala > mintz_potentzial_atalasea orduan
    For lotura_sinaptikoa: neuronaren irteerako lotura sinaptikoak egin
      lotura_sinaptikoa → gehitu_pultsoa = t + lotura_sinaptikoa → latentzia
    Amaiera For
    neurona → mintz_potentziala = deskantsuko_mintz_potentziala
    neurona → deskantsuan = 1
  Amaiera Baldin

```

1. Algoritmoan kodea hainbat begiztatan banatzen da, baina abantaila gisa paralelizagarria da PUZe an eta GPUan. Exekuzio-denborak eta lortutako azelerazioak 4. atalean aztertzen dira. Neuronen prozesamendurako bi begiztak banandu izanaren arrazoia (guztia begizta batean egitea posiblea izanik) kodea paralelizatzean sortutako arazoak ekiditea da. Neurona batentzat sarrerako lotura sinaptikoa dena, beste batentzat irteerako lotura da, eta ondorioz, baldintza konketu batzuk ematen direnean lasterketa baldintzak sortu daitezke kodea paraleloan exekutatzean emaitza okerrak sortuz. Banaketak arazo hori saihesten du.

3.4 Simulazioaren memoria okupazio estimazioa

Simulazioan garrantzitsua da behar den memoria kopurua jakitea. Sarea hasieratuta dagoenean (neuronen eta loturen zerrendak sortu eta hauen aldagai guztien balioak ezarrita), memoriaren okupazioa neurona eta lotura kopuruen arabera da. Sareari dagokion egiturak bi zerrenda eta hainbat aldagai ditu, baina bi zerrenden tamaina dela eta, gainerako aldagaiak guztiz baztergarriak dira. Neuronek behar duten memoria kopurua neurona motaren arabera da. Oraingoz LIF neuronak bakarrik inplementatu direnez, estimazioa egiteko neurona mota horiek behar duten memoria erabili da. Neurona bakoitzak hainbat zenbaki oso eta erreal ditu, guztira 14 zenbaki, beraz 56 byte. Bestalde, neurona bakoitzak sarrerako eta irteerako loturen zerrendak ditu. Lotura bakoitzaren indizea bi aldiz agertzen denez, N neurona eta M lotura izanik, neuronen zerrendak behar duen memoria kopurua $56N + 2M * 4 = 56N + 8M$ bytekoa da.

Lotura sinaptiko bakoitzari dagokionez, berriz, alde batetik kontrolerako aldagaiak dituzte, guztira 44 byte, eta, bestetik, lotura bakoitzak zerrenda zirkular bat du, pultsuak prozesatu beharreko uneak gordetzeko. Zerrenda horren luzera zehaztu beharreko kontua da, eta eragin handia du beharrezko memorian. Zerrendaren luzera L eta lotura sinaptiko kopurua M izanik, lotura sinaptikoei behar duten memoria kopurua $(44 + 4L)M$ bytekoa da.

Guztira, neurona-sarea sortu ostean, simulazioan zehar programak behar duen memoria kopurua $56N + (52 + 4L)M$ bytekoa da.

4 Exekuzio-denborak eta azelerazio-faktoreak

Atal honetan, tamaina desberdinetako bi sarerekin gauzatutako simulazioen emaitzak aurkeztuko dira: exekuzio-denborak, azelerazio-faktoreak eta lortutako eraginkortasunak. Simulazioetan erabilitako sareetan, lotura sinaptikoen kopurua neurona kopurua baino askoz gehiago hazten da. Erabaki hau hartu da neurona-sare biologikoetan neurona bakoitzak milaka lotura izan ditzakeelako.

Bestalde, esperimentazioan erabilitako sareen topologiak guztiz ausazkoak dira. Sarrerako pulsu sekuentziak ere ausazkoak dira. Exekuzioak gauzatzeko erabilitako hardwareari dagokionez, prozesadorea...

Bi kasuetan 10.000 iterazioko simulazioak gauzatu dira (simulazio luzeak) eta emaitzak 2 eta 3 Tauletan daude. Tauletan, prozesadore kopuru bakoitzarekin guztira lortutako exekuzio-denborak agertzen dira, baina, horretaz gain, neuronen eta ikaskuntza-erregelen prozesamendurekin lotutako atalen exekuzio-denborak bananduta aurki daitezke zati bakoitzaren eragina aztertzeko. Azelerazio-faktore eta eraginkortasunak 2. Irudian ere islatzen dira grafikoki, (a) Irudian 2. Taulakoak, eta (b) Irudian 3. Taulakoak.

Bi kasuetan, azelerazio-faktoreek paralelismoaren eragina islatzen dute, sare handien simulazioak gauzatzea ahalbidetuz. Gainera, eskalagarritasun ahula islatzen da, hau da, sarea handiagoa denean (simulazio handiagoa) paralelismo maila altuagoak lortzen dira. Hala ere, nabarmendu beharra dago emaitzak ez direla izugarri onak. Horren arrazoi nagusia cache memoriaren erabilera izan daiteke. Lotura sinaptiko guztiak zerrenda batean antolatuta daude, eta beraz, neuronak paraleloan prozesatzean, baliteke lotura sinaptiko berdinak prozesu desberdinek atzitzea, koherentzia arazoak sortuz eta exekuzioa motelduz.

Bestalde, simulazioen artean exekuzio-denboren hazkundeak lotura sinaptiko kopuruarekin neurona kopuruarekin baino erlazio handiagoa duela egiaztatu daiteke. Kontuan izanik neurona bakoitza prozesatzean lotura sinaptikoak zeharkatu behar direla (ikus 2 eta 3. Algoritmoak), argi dago hauen kopuruak eragin handia duela exekuzio-denboran.

Azkenik, sareen tamaina handitzen denean ikaskuntza-erregelen konputazioari dagokion atalak garrantzia galzten du exekuzioan, neuronen prozesamendua izanik denbora gehien behar duen atala.

5 Ondorioak

PNSen inguruko ikerkuntzaren hazkundera dela eta, software eta hardware tresnak ezinbestekoak dira hauen garapena bultzatzeko. Nahiz eta software tresna ugari dauden, asko garapenean edo erabilera zehatzetara mugatuta daude. Ondorioz, liburutegi edo simulatzaile orokorrako bat, sareak sortu eta erabiltzeko askatasun handia ematen duena lagungarria izan daiteke testuinguru batzuetan. Horretaz gain, C programazio-lengoaian inplementatzeak errendimendu egokiak lortzeko aukera berma dezake.

Bestalde, 4. atalean aurkeztutako emaitzek argi islatzen dute paralelismoak izan dezakeen eragina sare handien simulazioa gauzatzeko orduan, eta liburutegiaren erabilgarritasuna sare handiekin. Azkenik, aipatu beharra dago

2. Taula: 10.000 neurona eta 5.000.000 lotura sinaptiko dituen sarearen emaitzak 10.000 denbora-pausotan.

Prozesu kop.	Exek. denb. (s)	Neuronen exe. denb. (s)	IE exek. denb. (s)	AF	AF neuronak	AF IE	Erag.
1	4072.36	3089.77	982.58	1.00	1.00	1.00	1.00
2	3131.33	1941.99	1189.32	1.30	1.59	0.83	0.65
4	1542.4	946.82	595.55	2.64	3.26	1.65	0.66
8	779.54	496.25	283.25	5.22	6.23	3.47	0.65
16	483.44	331.66	171.69	8.42	9.91	5.72	0.53
24	369.79	227.24	142.45	11.01	13.60	6.90	0.46
32	349.38	201.47	147.81	11.66	15.34	6.65	0.36
48	278.41	164.58	113.62	14.63	18.77	8.65	0.30
64	269.46	150.35	118.82	15.11	20.55	8.27	0.24
96	260.66	135.82	123.88	15.62	22.75	7.93	0.16

3. Taula: 25.000 neurona eta 30.000.000 lotura sinaptiko dituen sarearen emaitzak 10.000 denbora-pausotan.

Prozesu kop.	Exek. denb. (s)	Neuronen exe. denb. (s)	IE exek. denb. (s)	AF	AF neuronak	AF IE	Erag.
1	26452.63	20911.13	5541.5	1.00	1.00	1.00	1.00
2	17469.13	10876.65	6592.45	1.51	1.92	0.84	0.76
4	10047.96	6775.83	3272.09	2.63	3.09	1.69	0.66
8	6097.53	4015.87	2081.54	4.34	5.21	2.66	0.54
16	3064.29	1959.51	1104.58	8.63	10.67	5.02	0.54
24	2246.07	1433.76	812.05	11.78	14.58	6.82	0.49
32	1838.76	1204.4	634.02	14.39	17.36	8.74	0.45
48	1345.65	907.37	437.7	19.66	23.05	12.66	0.41
64	1142.89	795.21	346.07	23.15	26.30	16.01	0.36
96	939.87	679.95	250.23	28.14	30.75	22.15	0.29

simulatzailea oraindik lehenengo bertsioan dagoela, eta, beraz, kontu asko daudela inplementatzeko eta hobetzeko, 6. atalean azalduko den moduan.

6 Etorkizunerako planteatzen den norabidea

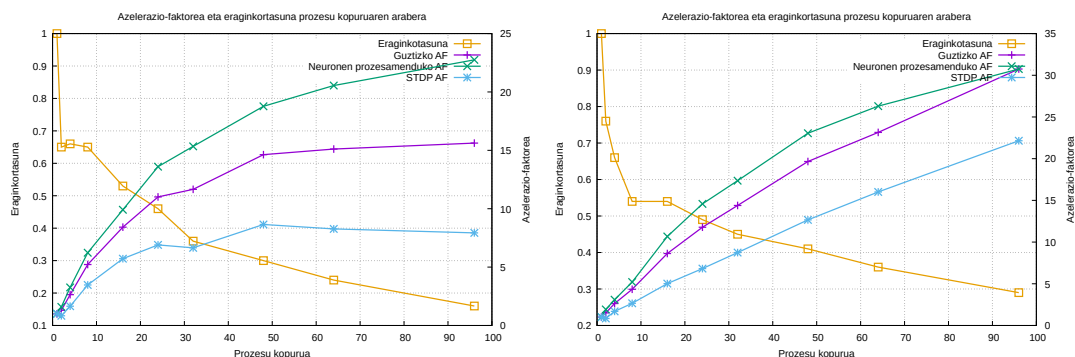
Hainbat hedapen eta gehikuntza garrantzitsu ezinbestekoak dira liburutegian. Alde batetik, simulazioetan neurona mota gehiago erabili ahal izatea beharrezkoa da, beraz, neurona eredu gehiago inplementatu behar dira.

Bestalde, kodean optimizazioak eta GPU bidezko azelerazioa ezinbestekoak dira sare handien simulazioak gauzatzeko. OpenMP erabiliz azelerazio maila egokiak lor daitezke, baina sare oso handiak simulatzean GPUak izugarritzko azelerazioa lor dezake PUZarekin alderatuta. Gertaeretan oinarritutako exekuzio-eredua inplementatzea ere baliagarria izan daiteke, ikaskuntza-erregela gehiago inplementatzeaz gain.

Azkenik, liburutegi hau sare biologikoetan oinarritutako topologiak aurkitzeko algoritmo genetiko baten inple-

2. Irudia

(a) 10.000 neurona eta 5.000.000 lotura sinaptikoko sarearen si-(b) 25.000 neurona eta 30.000.000 lotura sinaptikoko sarearen mulazioko azelerazio-faktore eta eraginkortasunak. simulazioko azelerazio-faktore eta eraginkortasunak.



mentazioan erabiliko da. PNSetan sare topologiak oso askeak izan daitezke. Gainera, lotura sinaptikoek latentziak izan ditzakete eta entrenamendu-erregela desberdinak erabili daitezke sare berean. Horrelako sare arbitrarioak eskuz diseinatzea konplexua da, bilaketa espazioa oso zabala baita. Ondorioz, NAS teknikak oso baliagarriak dira arkitektura egokiak aurkitzeko automatikoki. Neurona-sare biologikoetan aurkitzen diren neuronen zirkuituak oinarritzko elementutzat erabiliz, helburua ikaskuntza automatikorako oinarri biologiko sendoak dituzten topologia egokiak aurkitzea da, entrenamendu gunen desberdinak, loturen latentziak eta topologia askeak kontuan izanik bilaketa espazioan.

Erreferentziak

- Archana, R., eta P. S. Eliahim Jeevaraj. 2024. Deep learning models for digital image processing: a review. *Artificial Intelligence Review* 57.
- Cox, David Daniel, eta Thomas Dean, 2014. Neural networks and neuroscience-inspired computer vision.
- Gerstner, Wulfram, eta Werner M. Kistler. 2002. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press.
- Hunsberger, Eric, eta Chris Eliasmith, 2015. Spiking deep networks with lif neurons.
- Mittal, Divyansh, eta Rishikesh Narayanan, 2024. Network motifs in cellular neurophysiology.
- Otter, Daniel W., Julian R. Medina, eta Jugal K. Kalita. 2021. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems* 32.604–624.
- Pan, Wenxuan, Feifei Zhao, Zhuoya Zhao, eta Yi Zeng, 2023. Brain-inspired evolutionary architectures for spiking neural networks.
- Parker, Julian D., Janne Spijkervet, Katerina Kosta, Furkan Yesiler, Boris Kuznetsov, Ju-Chiang Wang, Matt Avent, Jitong Chen, eta Duc Le. 2024. Stemgen: A music generation model that listens. 1116–1120. Institute of Electrical and Electronics Engineers (IEEE).
- Perez-Nieves, Nicolas, eta Dan F. M. Goodman. 2021. Sparse spiking gradient descent.
- Shen, Shuaijie, Rui Zhang, Chao Wang, Renzhuo Huang, Aiersi Tuerhong, Qinghai Guo, Zhichao Lu, Jianguo Zhang, eta Luziwei Leng. 2024. Evolutionary spiking neural networks: A survey. Technical report.
- Xu, Yinlena, Silverio Martínez-Fernández, Matias Martinez, eta Xavier Franch. 2023. Energy efficiency of training neural network architectures: An empirical study.

7 Eskerrak eta oharrak

Ikerketa-lan hau Eusko Jaurlaritzak finantzatu du: KK-2023/00012, KK-2023/00090, KK-2024/00030, KK-2024/00068 eta IT1504-22 proiektuen bidez. Halaber, MICIU/AEI/10.13039/501100011033 erakundeak finantzatutako CNS2023-144315 dirulaguntzaren bidez eta "European Union NextGenerationEU/PRTR" programaren bidez. Era berean, PID2023-152390NB-I00 dirulaguntza MICIU/AEI/10.13039/501100011033 erakundeak finantzatuta, "FEDER funtsen" bidez. Dr. Javier Navaridas-ek Ramón y Cajal beka bat du, Espainiako Zientzia, Berrikuntza eta Unibertsitate Ministerioak (MCIN/AEI/10.13039/501100011033) finantzatua, eta, kasuan kasu, "ESF Investing in your future" edo "European Union NextGenerationEU/PRTR" programaren bidez babestua.