

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

Tesiaren egilea

Adrián Teso Fernández de Betoño

Tesiaren zuzendaria

Ekaitz Zulueta Guerrero

Doktoretza programa

Kontrol Ingeniaritza, Automatizazio ta Robotika

Aurkezpen urtea

2023



VITORIA-GASTEIZKO
INGENIARITZA
ESKOLA
ESCUELA
DE INGENIERÍA
DE VITORIA-GASTEIZ

Laburpena

Tesi honek sare neuronalen ikasketa prozesuaren ikerketa eta horri esker hobekuntzak planteatzea du helburu bezala. Muga horretara heltzeko, sare neuronalak ekoizteko eta ikuskatzeko ahalmena duen Software liburutegi bat eraiki egin da. Liburutegi honen beharra, normalean erabiltzen diren sare neuronalak sortzeko liburutegiek ez dituztelako eskaintzen beharrezkoak diren aldagaiak entrenamendu prozedura bera ulertzeko eta hobekuntzak planteatzeko. Garatu den liburutegiari ezker, gradiente beherakor estokastikoko algoritmoa, sare neuronalen ikasketa algoritmo klasiko bat, ikertzea eta hobekuntzak planteatzera heldu da.

Abstract

The aim of this PhD Thesis is the reasearch on the the learning process of the Artificial Neural Networks, and through this research, to propose different kind of improvements on that learning process. To achieve such goal, a Software library has been developed, which can create and inspect artifical neural networks. The need of this kind of library comes from the lack of training variables, critical for the learning process inspection and to propose modifications, in the most popular Neural Networks creation libraries. Thanks to the develop library, some modifcaions has been proposed to the classical Neural Network learning algorithm, the Stochastic Gradient Descendant algorithm.

Aurkibidea

I. Sarrera	- 1 -
<hr/>	
1. Arrazioak eta Helburuak	- 1 -
2. Objektuetara zuzendutako programaketa	- 3 -
2.1. Ezaugarri nagusiak.....	- 3 -
2.2. Unified modeling language(UML)	- 4 -
2.3. Programaketa garbia	- 5 -
3. Sare neuronalen oinarriko kontzeptuak	- 6 -
3.1. Ikasketa ikuskatuta eta gradiente estokastiko beherakorra	- 10 -
3.2. Sorospen bidezko ikasketa eta algoritmo genetikoak.....	- 15 -
3.3. Sare neuronalek gainditu beharreko oztopoak.....	- 20 -
3.4. Sare neuronalen optimizaketak	- 22 -
3.5. Sare neuronalen aplikazio adibidea: Nabigazio autonomoa.....	- 26 -
II. Teknikaren egoera	- 30 -
<hr/>	
1. Sare neuronalen optimizaketa.....	- 30 -
2. Sare neuronalen ulergarritasuna.....	- 33 -
3. SGD Algoritmoaren hobekuntzak.....	- 35 -
III. BasqueNet liburutegiaren garapena	- 38 -
<hr/>	
1. Lortu nahi ziren objektiboak	- 38 -
1.1. Baldintza funtzionalak	- 39 -
1.2. Baldintza ez-funtzionalak	- 40 -
2. Liburutegiaren arkitektura	- 41 -
3. Detaileko diseinua.....	- 42 -
3.1. Layer familia	- 42 -
3.2. Ikasketa funtzioen familia	- 45 -
3.3. Ikasketa datuen klasea	- 46 -
3.4. Sarearen ikertzailearen klasea	- 46 -
3.5. BasqueNet klasea	- 47 -
4. Liburutegiarekin lan egiteko era	- 48 -

IV. Gradiente Estokastiko Beherakorreko ikasketaren problemak - 57 -

- 1. Elikadura zuzeneko sareak eta beraien gradiente estokastiko beherakorreko entrenamendua..... - 57 -**
- 2. Ikasketa algoritmoak: Aurkitutako problemak eta beraien analisia ... - 59 -**
- 3. Gradiente desagerketaren problema..... - 60 -**
- 4. Gradientearen aurkako zentzua - 62 -**
- 5. Garatutako Matlab liburutegiaren laguntzaz egindako frogaketak..... - 63 -**
 - 5.1. 1. Esperimentua, Mexikar kapela - 64 -
 - 5.2. 2. Esperimentua, Ackley Funtzioa. - 73 -
 - 5.3. 3. Esperimentua, Cross-In-Tray funtzioa - 75 -
 - 5.4. 4. Esperimentua, 5. De Jong funtzioa - 78 -

V. Proposatutako aldaketak Gradiente Estokastiko Beherakorraren ikasketan- 80 -

- 1. Ikasketa ratioa eta Dropout ratioa: Aurkitutako problemak, analisia eta aurkitutako irtenbideak - 81 -**
 - 1.1 Alpha dinamikoa..... - 81 -
 - 1.2 Dropout dinamikoa - 82 -
- 2. Egindako esperimentuak eta beraien emaitzak..... - 83 -**
 - 2.1. 1. Esperimentua, Mexikar kapela - 83 -
 - 2.2. 2. Esperimentua, Ackley funtzioa - 89 -
 - 2.3 3. Esperimentua, Cross-In-Tray funtzioa - 91 -
 - 2.4 4. Esperimentua Dropwave funtzioa - 94 -

VI. Ekarpenak, Ondorioak eta Hurrengo pausuak.....- 97 -

- 1. BasqueNet liburutegia - 97 -**
- 2. Ikerkuntzako emaitza - 98 -**
- 3. Lanaren hedatzea - 99 -**
- 4. Hurrengo pausuak..... - 100 -**

VII. Bibliografia- 101 -

Eranskinak- 119 -

1. BaskeNet fitxategia.....	- 119 -
2. AnalizaCapas fitxategia.....	- 129 -
3. Layer fitxategia	- 131 -
4. FullyConnected fitxategia.....	- 132 -
5. ProcessingLayer fitxategia.....	- 135 -
6. ReluLayer fitxategia	- 135 -
7. SigmoidLayer fitxategia	- 136 -
8. SoftMaxLayer fitxategia.....	- 136 -
9. Ikasketak fitxategia	- 137 -
10. IkasketaFuntzioa fitxategia	- 138 -
11. MSEIkasketa fitxategia	- 138 -
12. EntropiaGurtzatutakoIkasketa fitxategia.....	- 139 -

Irudien Aurkibidea

Irudia 1: Agile Manifestua eta sinatzaileak.	- 5 -
Irudia 2: Sare neuronal erraz baten eskema.	- 6 -
Irudia 3: Sare konboluzional baten eskema[94].	- 8 -
Irudia 4: Sare baten ikasketa prozesua[58].	- 10 -
Irudia 5: Sare neuronalaren errorearen harana.	- 12 -
Irudia 6: Algoritmoa maldan behera bidaia egin du.	- 12 -
Irudia 7: Algoritmoa trabatuta geratzen da minimo lokal batean.	- 13 -
Irudia 8: Algoritmoa, inertiari esker, minimo absolutura doa.	- 14 -
Irudia 9: Galera katastrofikoa erakusten duen algoritmo baten entrenamendua[72].	- 16 -
Irudia 10: NEAT algoritmoko mutazio erak[175].	- 17 -
Irudia 11: Espezie ezberdinen bilakaera belaunaldiz, belaunaldi.[157].	- 18 -
Irudia 12: Sare neuronal bat entrenatzeko beharrezko diren patroien bilaketaren eskema[180].	- 21 -
Irudia 13: RELU(urdinez), leakyRELU(gorritz) eta ELU(laranja) funtzioen forma.	- 23 -
Irudia 14: Momentu normalarekin egingo lukeen bidea.	- 24 -
Irudia 15: AdaGrad, RMSProp edo Adam-ekin lortu nahi den bidea.	- 25 -
Irudia 16: SLAM-en lan egiteko eraren eskema[247].	- 26 -
Irudia 17: Hondar sarea, sare normalarekin alderatuta[94]	- 27 -
Irudia 18: Adibide irudi bat non urdinez zoru bezala klasifikatu diren pixelak daude eta laranja/gorritz paretak[94].	- 28 -
Irudia 19: Irudia 17-tik ateratzen den bidea egiteko funtzioa[94].	- 28 -
Irudia 20: Zinta markatzailearen gainean kalkulaturako ibilbidea[93].	- 29 -
Irudia 21: Proposatutako nabigazio mapa[60].	- 29 -
Irudia 22: Auto-Keras-en prozedura grafikoa[260].	- 32 -
Irudia 23: Erabilera kasuak.	- 38 -
Irudia 24: Liburutegiaren arkitektura.	- 41 -
Irudia 25: Layer klase familia.	- 44 -
Irudia 26: Ikasketa funtzioen klaseen familia.	- 46 -
Irudia 27: Ikasketak klasea.	- 46 -
Irudia 28: AnalizaCapas klasea.	- 47 -
Irudia 29: BasqueNet klasea.	- 48 -
Irudia 30: Ikasketa patroien sorkuntza.	- 49 -
Irudia 31: Ikasi nahi den funtzioaren forma.	- 49 -
Irudia 32: Sarearen sorkuntza.	- 50 -
Irudia 33: Entrenamendu scripta.	- 50 -
Irudia 34: Entrenamenduko grafikak.	- 50 -
Irudia 35: Lortutako forma eta errorea.	- 51 -
Irudia 36: Loss-aren bilakaera epokaz-epoka.	- 51 -
Irudia 37: Patroiz-patroi geruza bakoitzaren modulua azkeneko epokan.	- 52 -
Irudia 38: Sarrerako geruzan gordetzen diren datuak.	- 53 -
Irudia 39: Sare ikertzaile barruan gordetzen diren patroien datuak.	- 53 -
Irudia 40: Azkeneko epokako lehengo patroiak sortu dituen aldaketak geruzaz-geruza eta bakoitzak jaso duen sarrera.	- 53 -
Irudia 41: BasqueNeteko barruko datuak.	- 54 -
Irudia 42: Txartoen ikasi den patroien datuak.	- 56 -

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

Irudia 43: Txartoen ikasi diren 20 patroien batezbesteko jokaera.	- 56 -
Irudia 44: Sare neuronal baten irudikapen orokorra.	- 60 -
Irudia 45: (8) ekuazioen elementuen tamainaren irudikapena.	- 60 -
Irudia 46: Ikerketarako erabilitako sarea.	- 61 -
Irudia 47: Sortutako sare neuronalaren deskribapena.	- 63 -
Irudia 48: Entrenamenduaren emaitzak.	- 64 -
Irudia 49: Epokaz-epoka batezbesteko Loss balorea.	- 64 -
Irudia 50: Txartoen ikasi diren 20 patroien zuzenketa kosinua ezkerreko zutabea eta $\partial \text{Loss}_{\mu \theta \text{Wn}}$ -ren moduluaren erlazioa batezbestekoarekiko eskuman.	- 68 -
Irudia 51: Hoberen ikasi diren patroien zuzenketak sarerako, ezkerreko zutabea kosinua edukirik eta eskumakoan $\partial \text{Loss}_{\mu \theta \text{Wn}}$ -ren moduluaren erlazioa batezbestekoarekiko.	- 69 -
Irudia 52: Txartoen ikasi diren patroien zuzenketak sarerako, era bildu baten. Ezkerreko grafikari kosinua egonik eta eskuman $\partial \text{Loss}_{\mu \theta \text{Wn}}$ -ren moduluaren erlazioa batezbestekoarekiko.	- 70 -
Irudia 53: Hoberen ikasi diren patroien zuzenketak sarerako era Bildu baten. Ezkerreko grafikari kosinua egonik eta eskuman $\partial \text{Loss}_{\mu \theta \text{Wn}}$ -ren moduluaren erlazioa batezbestekoarekiko.	- 70 -
Irudia 54: Patroien banaketa guztiz konektaturik dauden geruzetan, errore erlatibo eta heina erlatiboaren arabera.	- 71 -
Irudia 55: Patroien banaketa, errore erlatibo eta heina erlatiboaren arabera X_2 geruzan. ...	- 71 -
Irudia 56: Patroien banaketa, errore erlatibo eta heina erlatiboaren arabera X_4 geruzan. ...	- 72 -
Irudia 57: Batezbesteko errorea heina erlatiboarekiko. X_2 ezkerrean eta X_4 eskuman.	- 72 -
Irudia 58: Heina erlatiboaren eta errore erlatiboaren arteko erregresio polinomikoa X_2 ezkerrean eta X_4 eskuman.	- 73 -
Irudia 59: Ackley funtzioaren entrenamendu emaitzak.	- 73 -
Irudia 60: Lossaren batezbestekoaren balioa epokaz-epoka.	- 74 -
Irudia 61: Txartoen ikasitako patroien kosinua ezkerrean eta batezbestekoarekiko modulu erlazioa eskuman.	- 74 -
Irudia 62: Hoberen ikasitako patroien kosinua ezkerrean eta batezbestekoarekiko modulu erlazioa eskuman.	- 74 -
Irudia 63: Erregresio polinomikoa heina erlatiboa errore erlatiboarekiko. Ezkerrean X_2 egonda eta eskuman X_4	- 75 -
Irudia 64: Cross-In-Tray funtzioaren emaitzak.	- 75 -
Irudia 65: Lossaren batezbestekoaren balioa epokaz-epoka.	- 76 -
Irudia 66: Txartoen ikasi diren patroien zuzenketen angeluaren kosinua ezkerrean eta moduluaren erlazioa eskuman.	- 76 -
Irudia 67: Hoberen ikasi diren patroien kosinu erlazioa ezkerrean eta moduluaren erlazioa eskuman.	- 76 -
Irudia 68: heina eta errore erlatiboaren arteko erregresio polinomikoa. X_2 ezkerrean eta X_4 eskuman.	- 77 -
Irudia 69: De-Jong funtzioaren emaitzak.	- 78 -
Irudia 70: Lossaren batezbestekoaren balioa epokaz-epoka.	- 78 -
Irudia 71: Txartoen ikasi diren patroien kosinua ezkerrean eta moduluen erlazioa eskuman. -	- 79 -
Irudia 72: Hoberen ikasi diren patroien kosinua ezkerrean eta moduluen erlazioa eskuman. -	- 79 -
Irudia 73: X_2 -n ezkerrean eta X_4 -n eskuman sortzen den Heina/errore erlatiboko erregresio linealak.	- 79 -

Irudia 74: Batezbesteko Loss funtzioaren balioa, urdina entrenamendu patroiena izanik eta laranja balidazio patroiena.....	83 -
Irudia 75: Batezbesteko Loss funtzioaren balioa alpha dinamikoa aplikatuta, urdina entrenamendu patroiena izanik eta laranja balidazio patroiena.....	84 -
Irudia 76: Alpha dinamikoa aplikatuta, txartoen ikasi diren patroien $\cos(\varphi)$ ezkerrean eta moduluen erlazioa eskuman.	84 -
Irudia 77: Alpha dinamikoa aplikatuta, hoberen ikasi diren patroien $\cos(\varphi)$ ezkerrean eta moduluen erlazioa eskuman.	85 -
Irudia 78: Batezbesteko Loss-aren balioa entrenamendu patroietan urdinez eta balidaziokoetan laranja.	85 -
Irudia 79: Txartoen ikasi diren patroien datuak, ezkerrean kosinua eta eskuman moduluen erlazioa.	86 -
Irudia 80: Hoberen ikasi diren patroien datuak, ezkerrean kosinua eta eskuman moduluen erlazioa.	86 -
Irudia 81: Batezbesteko Lossaren balioa, usdinez entrenamendukoa eta laranja balidaziokoa..	87 -
Irudia 82: Ezkerrean alpha dinamikoa bakarrik aplikatuta lortzen diren erroreak patroiekiko eta eskuman bi aldaketak konbinaturik lortzen direnak.....	87 -
Irudia 83: Bi aldaketak batera aplikaturik, txartoen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.....	88 -
Irudia 84: : Bi aldaketak batera aplikaturik, hoberen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.....	88 -
Irudia 85: Batezbesteko Loss-a Entrenamendu patroietan urdinez eta balidaziokoetan laranja.	89 -
Irudia 86: Txartoen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.	89 -
Irudia 87: Hoberen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.	90 -
Irudia 88: Batezbesteko Loss-a Entrenamendu patroietan urdinez eta balidaziokoetan laranja.	90 -
Irudia 89: Txartoen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.	91 -
Irudia 90: Hoberen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.	91 -
Irudia 91: Batezbesteko Loss-a Entrenamendu patroietan urdinez eta balidaziokoetan laranja.	91 -
Irudia 92: Batezbesteko Loss-a Entrenamendu patroietan urdinez eta balidaziokoetan laranja.	92 -
Irudia 93: Txartoen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.	92 -
Irudia 94: Hoberen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.	93 -
Irudia 95: Txartoen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.	93 -
Irudia 96: Hoberen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.	93 -
Irudia 97: SGD normalarekin batezbesteko Loss-a Entrenamendu patroietan urdinez eta balidaziokoetan laranja.	94 -

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

Irudia 98: : Aldaketak egin ostean batezbesteko Loss-a Entrenamendu patroietan urdinez eta balidaziokoetan laranja.	- 94 -
Irudia 99: Txartoen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.	- 95 -
Irudia 100: Hoberen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.	- 95 -
Irudia 101: : Txartoen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.	- 96 -
Irudia 102: Hoberen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.	- 96 -
Irudia 103: Databeers seminarioan egindako aurkezpena.	- 99 -

Taulen Aurkibidea

Taula 1: Errore haundiena duen patroia, geruzaz geruza batezbestekotik alderatzen duen angeluaren kosinua eta moduluen erlazioa..... - 65 -

Taula 2: Errore txikiena duen patroia, geruzaz geruza batezbestekotik alderatzen duen angeluaren kosinua eta moduluen erlazioa..... - 66 -

Taula 3: Errore haundiena duen patroia geruzen zuzenketaren kosinua ta modulu erlazioa batezbestekoarekiko..... - 66 -

Taula 4: Errore txikiena duen patroia geruzen zuzenketaren kosinua eta modulu erlazioa batezbestekoarekiko..... - 67 -

Glosategia eta Laburdurak

Drop-out:	Sare Neuronalen ikasketa teknika, non zenbait pisu sinaptiko ikasketatik at geratzen diren, sarea era orokorrago baten ikas dezan.
Loss:	Sare neuronal batek duen errorea irudikatzen duen funtzioa. Honen deribatu partzialei ezker sarearen ikasketa zuzenketak plantatzen dira.
MSE:	Minimum Square Error. Errore karratu minimoa.
OZP:	Objektuetara zuzendutako programaketa.
ReLU:	Rectifier Linear Unit. Zuzenketa linealeko unitatea. Zuzenketa linealaren funtzioa aplikatzen duen unitatea.
SGD:	Stochastic Gradient Descendant. Gradiente Estokastiko Beherakorra.
UML:	Unified Modelling Language, modelaketa hizkuntza bateratua. Programaketaren diseinua eta baldintzak ezartzeko diagrametan oinarritutako lengoaia.

Aipamenak eta Esker-emateak

Doktoretza hau ezin lirateke izan nire lagunen eta familiaren euskarriagatik, alde teknikitik at, zuntza eta arima eman diote. Marta, nire emaztea, lehenik aipatu nahi dut, bere laguntza eta pazientziagatik; bai testuak idazten nituenean edota simulazioak prestatu behar nituenean, dena beltz ikusten zenean, bera bertan egon da argia emateko.

Bigarren, nire gainontzako familiari, batez ere anaia eta amari, beti bertan egoteagatik. Proiektuak ezin dira aurrera atera, atzean euskarri diren kideak izan barik, eta beraiek izan dira nire euskarri, bai ama, beti animuak ematen aurrera ateratzeko bai anaia, doktoretza haztearen erruduna dena, berak berea atera zuenean bertan aztarna bat utz egin zuen nerea ere aurrera ateratzeko. Gainontzeko Betoñadari ere milesker, familia bertan egon ezean ezin da honelako proiektu bat aurrera eraman.

Nire lagunei, Ninja Purpurei, Fiore dei Liberi, Spinky eta gainontzekoei ere. Milesker, garagardo bat hartzeko beti bertan egoteagatik, kontzertu on batera juteko, ezpatak gurutzatzeko prest egoteagatik, edota mahai baten inguruan, ziegak aztertzeko eta herensugeak akabatzeko ausart egoteagatik bazkari on baten ostean. Bihotz bihotzez milesker guztioi. Nahiz eta, azkeneko urte hauetan COVID madarikatuaren erruagatik ez garen hainbeste egon, beti buruan eduki zaituztet denak.

Azkenik Gasteizko Ingeniaritza eskolari eta nire tesi zuzendariari, Ekaitz Zulueta Guerrero, eman didaten aukeragatik. Ekaitzekin lan ordu luzeak izan ditugu berandu arte ideak lantzen, azkenean, eduki dugun lan ona ateratzeko kapaz izan gara.

I. Sarrera

1. Arrazioak eta Helburuak

Hasiera baten doktoretza honen helburua, eraikin barreneko ibilgailuentzako sare neuronal sakonetan oinarritutako algoritmo bat sortzea zen. Algoritmo honen helburua, errekurtsio minimoak erabilia, ibilgailua lan egiteko gaitasuna izatea zen. Algoritmo hau sortzeko teknika ezberdinak ikertu ziren, alde batetik, hari bidez gidatutako estrategia, non, hari magnetiko bat eremuaren zoruan jartzen da, ibilgailua hauek jarraituta bideratu egiten den, eta bestalde, aldibereko mapatu eta bideraketa estrategia erabiltzea, non, ikusmen artifizialaren bitartez ibilgailua bideratzen da, paretak eta oztopoak saihestuz. Helburua zen, ingurumen industrial baten barruan aske ibil zezakeen gailu bat sortzea, bere posizio zehatza jakin barik lan egin zezakeena. Hau lortzeko, segmentazio semantiko[1], [2] bitartez lan egiten duten sare neuronal konbolutiboak[3]-[6] erabili nahi zen. Ere, NEAT[7] erako algoritmo neurobolutiboak aztertzeko intentzioa zegoen, naiz eta zaharrak izan, bideojolasetan erabilitako bot-ak entrenatzeko arrakastatsua izan dira.

Ala ere, lan guzti hau garatzeko, beharrezko ikusten zen, algoritmoa ulergarria izateko eta beharrezkoak ziren errekurtsioak mugatzeko, sare neuronalak diseinatzeko software liburutegi bat sortzea. Liburutegia MATLAB garapen ingurumenean landu egin nahi zen, bertan lantzeko jakituria baitzegoen eta ingurumena bera datuak gorde eta lantzeko aukera ugari eskaintzen baititu. Erabili beharko zen programaketa objektuetara zuzenduta egon behar da, bere egituran aldaketak edo zabaltzeak onartu behar baititu. Era berdinean, programaketa hau era garbi baten planteatu behar da, sarearen diseinua ulergarria izan behar delako, aldaketak xamurrak izan daitezten. Liburutegia garatzen zen bitartean, lortzen ziren emaitza partzialak ikusirik, honen arrazoiak eta helburuak aldatzeko erabaki zen, liburutegia beraren garapena ta lortu ahal ziren emaitzak arrazoi bihurtuz. Liburutegi honetaz, sareak ikasteko erabiltzen diren datuak analizatzeko aukera ematen du eta hauek daukaten eragina sarearengan aztertu. Ere, sareen ikasketarako erabiltzen diren algoritmoen portaera eta jarduera aztertzen ahalbidetzea espero da.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

BasqueNet deitu den liburutegi honekin, helburua, sare neuronalen ikasketa analizatzea eta honen bitartez, hobekuntzak planteatzea izan da. Hobekuntza hauek, liburutegian bertan integratzeko ahalmena eduki beharko lukete eta behar beste datu gorde, bai ikasketaren zehar datu hauek aztertzeko, bai behin ikasketa bukatu ahala patroiek eta sarearen pisu sinaptikoek eduki duten portaera aztertu eta egin beharreko aldaketa planteatzeko ahalmena eduki beharko luke. Liburutegiarekin lan egiteko era hurrengoko hau izan behako luke:

- 1. Liburutegiaz, sare neuronala sortu eta erabiliko diren ikasketa parametroak ezarri.**
- 2. Entrenamendu eta balidazio patroiak definiturik sarea entrenatzera jarri.**
- 3. Entrenamenduan zehar, sarea edukitzen ari den portaeraren datuak hartu eta grafiketan islatu.**
- 4. Entrenatu ostean, lortu diren datuekin, sareak izan duen portaera ikertu eta ondorioak atera.**
- 5. Hobekuntzaren bat aurkitu ezkerro, liburutegian garatu eta berriz prozesua hasi.**

Hurrengo orrietan islatzen den moduan, gradiente estokastiko beherakorra hartu da eredu moduan eta honen azterketa egin ostean sare neuronal sakonetan, errekurtsio ugari eskatzen ez duten pare bat aldaketa proposatzera iritsi da tesi honen bilakaeran.

2. Objektuetara zuzendutako programaketa

Objektuetara zuzendutako programaketa (OZP) 1966-an planteatu izan zen Ole Johan Dahl eta Kristen Nygaard-engatik[8]. Gaur egun sortzen diren programaketa hizkuntza gehienak filosofia honetan (Python, C++, C#, Java, JS,...) daude oinarriturik[8]-[14]. Programaketa klasikoarekin konparatu ezker, datuak Klaseak deitzen diren esturktoretan definitzen dira; klase hauekin metodo eraikitzailea deitzen den funtzio baten bitartez memorian alokatzen dira, objektu beza ezagutzen dena. Objektu hauekin erlazionaturik dauden funtzioak metodo bezala ezagutzen dira.

2.1. Ezaugarri nagusiak

Klaseen eta objektuen erabilerak, interesgarriak diren efektuak dakartza programaketan, kodea garbituz eta algoritmo konplexuagoak erraz ebazteko posibilitatea emanez. Hauek dira ezaugarri horiek[8]:

- **Datu eta Funtzioen kapsulapeena:** Klaseak eraikitzeko orduan, beraien datu eta funtzio batzuk, klasetik kanpo dauden elementuei izkutatzen zaizkie, huen erabilera eta aldaketa bakarrik klase horretako objektuei mugatuz. Era honetan, definitu den eremutik kanpo dauden elementuek ezin dute nahi ez diren aldaketarik sortu, bakarrik esarri diren eremuetatik pasa behar baitute.
- **Oinordepeena:** Klaseak beste batzen oinorde definitu daitezke. Era honetan oinordeko klasea, klase originalak zituen datu eta metodoak edukiko ditu, eta, hauetaz aparte, beste datu eta metodo berriak sortu ditzake. Era berean, klase originalaren metodoak berridatzi ditzake klase oinordeak, beste prozedura batzuk ezarriz metodo horretarako. Era honetan, klase originalarekin sortutako objektu batetan metodoa deitzerakoan, prozedura originala lan egingo du eta oinorde klaseko objektu baten metodoak prozedura berriarekin lan egingo du.
- **Polimorfismoa:** Ezaugarri hau ulertzeko zaienetakoa da. Interfaze deitutako elementu batzuen bitartez; interfaze hauetan, metodo multzo baten goiburuak ezartzen dira, baina ez beraien inplementazioa. Klase batek interfaze hori inplementatzen duela esan dezake, metodo hauei gorputz bat emanez. Era honetan beste objektu batzuk interfaze honetako metodo bat erabiltzerakoan, ez dute zertan ezagutu behar nor inplementatzen duen interfaze hori, kasuz kasu tokatzen den klasea erabiliz. Hau erabilgarria da oso, batez ere goi maiako funtzionalitateak Hardwaretik gertu dauden beheko

funtzionalitateekin lan egin nahi dutenean. Goi maiako funtzionalitatea beti interfaze berdinerara deituko du eta behe mailako funtzionalitateak interfaze horri gorputza emanaz, bakoitzak Hardwarea operatzeko egin behar dituen lanak inplementatuz.

Ezaugarri hauek, programaketa hizkuntza tradizioaletan ere eman zintezkeen, baina era oso mugatu batean edo seguru eza zan era batean. OZP-tara zuzendutako hizkuntzek ezaugarri hauek era fidagarri baten egiteko aukera ematen dute.

2.2. Unified modeling language(UML)

OZP-rekin oso lotuta dagoen bezte ezaugarri bat, modeloak dira. Klaseak direla medio eta beraien polimorfismo/oinordepeena erabilita, modeloak sortzeko aukera ematen duten hizkuntzak dira. UML modelo hizkuntza, 1990ko hamarkadaren bukaeran sortutako errementa bat da[15]-[18]. Hau erabilita, programatzen hasi baino lehenagoko pausuetan, sistemak edukiko dituen ezaugarrien modelaketa egin daiteke. Hasierako pausuetan, azkeneko inplementaziotik urrun egonda, sortzen diren modeloen forma oso abstraktua izan daiteke, sistemak izango dituen goi maiako baldintza definiturik eta beraien artean loturik. Gero, V formako prozesua jarraitu ezker, sortutako modelo diagramak geroz eta zehatzagoak bihurtzen dira, azkeneko inplementaziotara irisi arte. Gero azterketa prozesuan, modelo hauek edukirik errazago da mahai bakoitzean test horiek konprobatzea.

UML-ren helburuen artean, programatzailea ez den pertsona batek bertan sortzen diren diagramak ulertzeko ahalmena izan behar du, horretarako formakuntza berezirik jaso gabe[17]-[20], era honetan proiektuen partaide diren pertsonak, baina Software produkzioan parte hartu ez dutenek, zer egingo den ulertu dezakete eta programatzaileen artean ere, lortu beharreko helburuak eta sortu den kodigoaren azalpena erraztu egiten du, gerora programa horren mantentzea erraztuz eta sortu daitezkeen problemak ulergarriagoak bueltatuz[21].

2.3. Programaketa garbia

Programaketa garbia, Robert C. Martin[8], [22]-[25] proposatutako programaketa metodo bat da, non programaren testetan oinarrituriko garapena egiten den. Metodo honen funtsa, kodea sortzeko partean, programa bat sortzerakoan era ordenatuan garatuta izatea eta kodearen elementu bakoitza azterturik egotea da. Metodoaren filosofia aztertuta, SOLID[8] metodoan eta Agile manifestuan[26] oinarriturik dago, non etengabeko hobekuntza eta bakoitzaren programaketa diziplina sustatu egiten diren. Irudia 1-n manifestua eta beraren sinatzaileak ikusi daitezke.

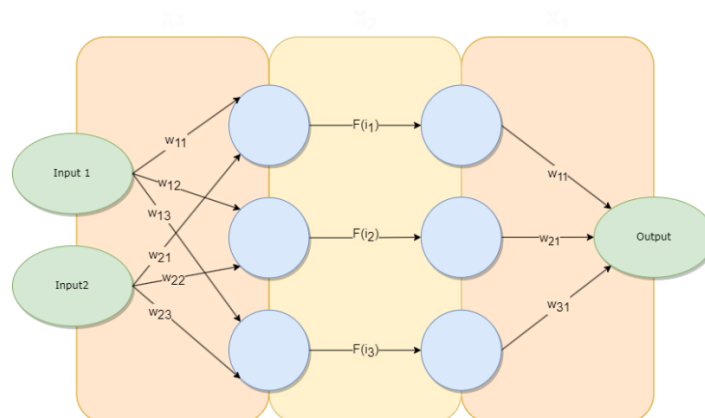


Irudia 1: Agile Manifestua eta sinatzaileak.

3. Sare neuronalen oinarriko kontzeptuak

Sare neuronalak, Machine Learning[27] barruan sartuta dagoen adimen artifizialeko algoritmoak dira, non burmuin baten geruza ezberdinak antzeratu nahi dira.[3], [28]. Machine Learning, 1959-n Arthur Samuelek definiturik[29], konputagailuek ikasteko ahalmena, zuzenki programaturik egon gabe, ikertzen duen arloa da[27]-[54]. Honek esan nahi du, konputagailu bat, adimen maila bat behar duen lan bat ikasteko ahalmena edukitzea, bere programaketatik kanpo; hau, programaketa klasikoarekin alderatzen du, non programaren portaera zuzenean programaturik daude[28]. Portaera honek, sarrera baten aurrean programak sortutako irteeraren ulergarritasuna ilundu dezake, problema bat izanik fidagarritasun haundi bat behar dituzten aplikazioetan[3], [55].

Sare hauen geruzak, neurona bezala ezagutzen diren nodulu batzuek osaturik daude. Nodulu bakoitza, geruzaren sarrera guztietara konektaturik dago, era ponderatu baten. Gero neurona bakoitza aktibazio funtzio bat du, neurona "pizteko" edo "itzaltzeko" ahalmena emanez[3], [28], [56]-[60]. Oraingo lanaren esparruan, geruzak bi mota ezberdinetan banandu dira, guztiz konektaturik dauden geruzak, ponderazio guztiak dituenak, eta prozedura geruzak, aktibazio funtzioa inplementatzen dutenak, era honetan, geroago egin beharreko kalkulu eta esperimentuen ikertzea erraztu egin da[56], [57]. Irudia 2-n ikusi daitezke nola antolatu egiten da sare neuronal erraz bat, sarreran eta irteeran guztiz konektaturik dauden geruzak egonda eta, erdialdean, prozedura geruza.



Irudia 2: Sare neuronal erraz baten eskema.

Gutziz konektaturiko geruzen ponderazioak, pisu sinaptikoak deritzenak, ikasketa prozesuan zehar modifikaturik izango dira, bai sarrera eta irteera erlazionatzen dituzten pisuak baita neurona bakoitza barnean daukan hasiera baloreak ere. Aktibazio funtzioaren aldetik, kasu bakoitzean beharrezkoa den funtzioa hartu daiteke. Hemen adibide batzuk:

→ **Zuzentzailea (ingelesez Rectified Linear Unit edo ReLU):** Mahai baten behetik dauden sarreraren igaropena ekiditen duen funtzioa da; mahaia hau normalean 0 izanik. Mahaia honetatik gora daudenak, era zuzen baten pasatzen uzten ditu[2].

→ **Sigmoidea:** Filtro moduan funtzionatzen duen funtzioa, sarrerako balore bortitzenak (0,1) eremuan sarturik[3], [28], [58]. Hurrengoko hau bere aktibazio funtzioa da:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

→ **Tangente hiperbolikoa:** Sigmoidea bezalako funtzio bat da, balore bortitzak filtratzen dituena, baina kasu honetan (-1,1) eremuan egonda.

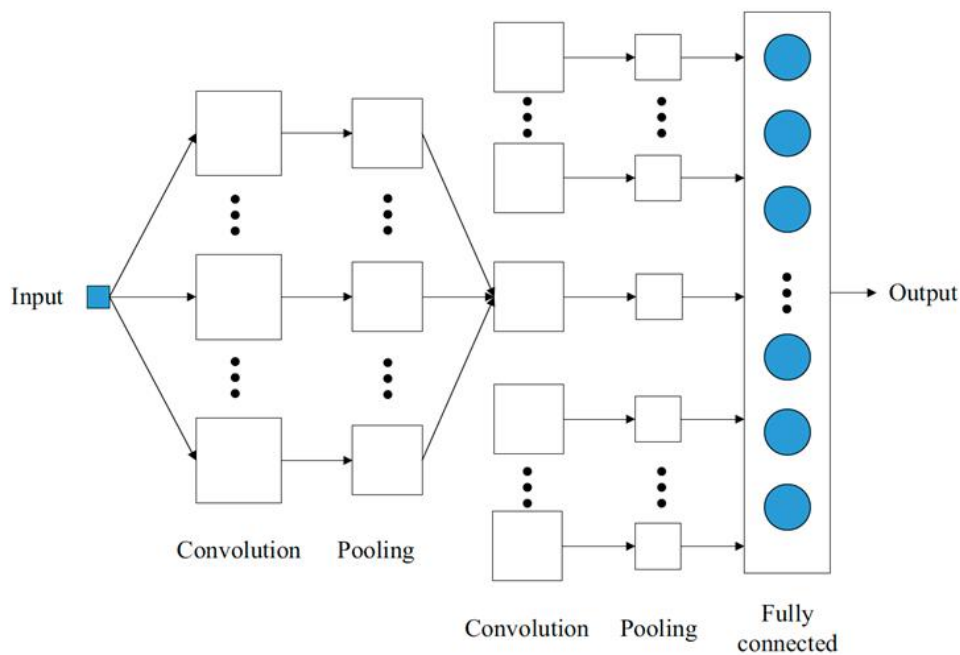
→ **Softmax:** Existitzen diren irteeren artean probabilitate distribuzio bat eskaintzen duen aktibazio funtzioa. Irteera hauek elarreko baztertzailerik izan behar dira.

Sarrera edo irteerarekin loturarik ez daukaten geruza ugari dituzten sareei, sare neuronal sakonak deritzate[3], [33], [61]-[73]. Sare hauek funtzio konplexuagoak ikasteko ahalmena dute kapa bakarreko edo kapa gutxiko sareak baino.

Sare neuronalen artean hiru multzo nagusi ezberdin daitezke[3], [73]:

→ **Geruza ugariko pertzeptroia edo elikadura zuzeneko sareak:** Sare sakonen artean sinpleenak dira. Klasifikapen edo erregresio ez linealak egiteko erabili egiten dira. Sare neuronal hauen inguruan landuko da tesi honen gorputz printzipala. Beraien neuronen matematika erraza izanez, pisu gutxiko sare neuronalak egiteko erabili daitezke beste motekin konparaturik.

→ Sare neuronal konboluziodunak: Sare neuronal hauek beste kapa mota bat integratzen dute, geruza konboluzioduna. Sare neuronal hauek irudien barruan ereduak aurkitzeko egokiak dira[5], [33], [37], [50], [63], [74]–[93]. Guztiz konektaturiko geruza baten eta konboluzio geruza baten arteko diferentzia funtzionala honako hau da, guztiz konektaturiko geruza batek patroien modelo orokor bat ikasteko ahalmena du, konboluzionalak, ordea, patroia txikiak ikasten ditu irudi orokor batetik abiturik[3]. Efektu hau behin eta berriz pilatu egiten bada geruza ezberdinak lerrokaturik, irudi batetik detaile oso zehatzak ikasteko ahalmena duten sareak sortu daitezke. Konboluzioaz aparte, sare neuronal hauek pooling deritzon operazio bat burutzen dute geruza konboluzional baten ostean. Operazio honetan, aurreko kaparen informazioaren sinplifikatze bat sortarazten da parametro bezala eman zaion irizpidea kontutan harturik. Irudia 3-n ikusi daiteke sare konboluzional baten eskema erraztua.



Irudia 3: Sare konboluzional baten eskema[94]

→ Sare neuronal errekursiboak: Sare mota hau, barne estatu bat gordetzen du, aurretik saretik pasa diren sarrerez elikatuta. Irteera, barneko estatu honen eta oraingo sarreraren menpe egongo da. Denborarekin edo testu prozedurarekin zerikusia duten aplikazioekin erabiltzen dira sare hauek[95]-[113].

Sare neuronal baten parte garrantzitsu bat, bere eraiketaz aparte, erabiliko den Loss funtzioaren aukeraketa da[114]-[121]. Funtzio hau, sarea entrenatzen dagoelarik entrenamendu elementu bakoitza pasa ostean sortu duen errorea, eduki beharko lukeen irteerarekiko errorea kalkulatzeko funtzioa da. Funtzio hau kalkulatu ostean, bere deribatu partzialaren bitartez, geruzaz, geruza pisu sinaptikoak zuzenduz. Batezbesteko errore karratua[122]-[128], oso erabilia den Loss funtzio bat da, erabilgarria batez ere elikadura zuzeneko sareetan, erregresio ez-lineal bat ikasi nahi denean. Oso erabilia den beste Loss funtzio mota bat, Gurutzatutako entropia da[129]-[138], errorea ondo sumatzen du, irteera anitzeko sareetan batez era, sailkapenerako funtzio egokia eginik.

Ikasketa motaren aldetik, 3.1, II, IV eta V puntuetan gradiente estokastikoari buruz luze hitz egingo da. Beste ikasketa moten artean[139]-[144], Levenberg-Marquard[145]-[155] algoritmoa aipatuko da. Karratu ez-lineal minimoen ebaztetarako erabiltzen den algoritmo hau gradiente estokastiko beherakorraren ordean erabili daiteke, bere ekuazioa hurrengo hau izanda:

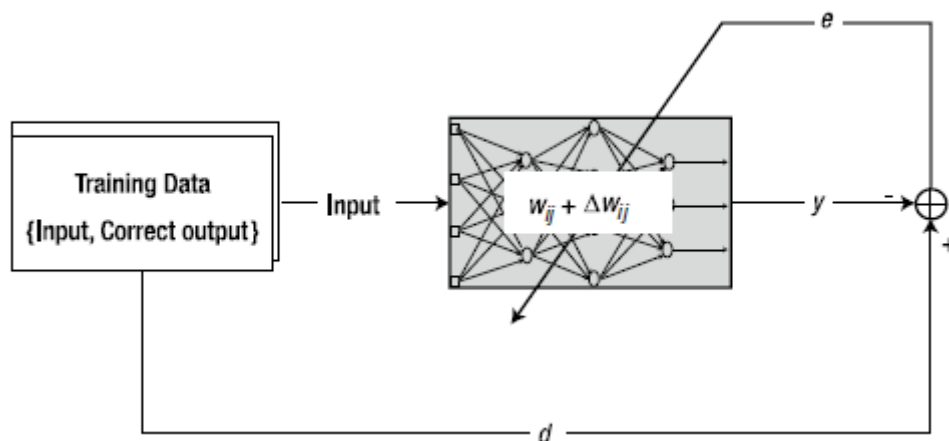
$$\nabla W_n = (J^T J + \lambda I)^{-1} \left[J^T \left(\frac{\delta \text{Loss}}{\delta X_n} \right) \right] \quad (2)$$

Non W_n n -garren geruzaren pisu sinaptikoen matrizea da, J sarearen pisu sinaptiko guztien errenkada bektorea da eta sartutako zuzenketa; bere balore minimoa 0 izanik, non karratu minimoen ebazte bat izango da eta geroz eta haundiagoa egiten den heinean, ebazte estokastiko komun bat bilakatzen da. I identitate matrize bat da J ren tamaina berberekoa. $\frac{\delta \text{Loss}}{\delta X_n}$ Loss funtzioaren deribatu partziala n -garren geruzarekiko islatzen du.

3.1. Ikasketa ikuskatuta eta gradiente estokastiko beherakorra

Hurrengo lerroetan, gradiente estokastiko beherakorreko ezaugarri nagusienak aurkeztuko dira. Lehenik eta behin, kontutan hartu behar da, sare neuronalek, era ikuskatu baten ikasten dutenean, atzerako hedaketaren bitartez ikasten dutela. Honek esan nahi du, ikasiko diren patroiak alde aurretik sortu beharko luketen irteerarekin markatu egiten dira, eta horri esker sareak sortzen duen errorea kalkula daiteke. Errore honetatik abiatuta, ikasketa algoritmoa sarean atzerantz pisu sinaptikoak zuzentzen ditu sarrerara heldu arte. Aurreko puntuan esan den moduan errorea kalkulatzeko funtzioari Loss funtzio bezala ezagutzen zaio eta ikasketa algoritmoarekin batera sarearen ikasketaren funtsa eraikitzen dute.

Irudia 4-n[58] ikusi daiteke era oso eskematiko batean nola den sare neuronal baten ikasketa prozedura era ikuskatuan egiten dutenean. Ezkerrean alde aurretik markaturiko patroiak daude, "Training Data", hauek sarearen sarrera elementuak izango dira eta saretik igaroko dira, "y" bezala markatutako irteera sortuz. Irteera hau patroiarekin datorren "d" esperotako irteerarekin alderatu egiten da Loss funtzioa aplikatuz eta horrekin "e" errorea sortuz. "e" honi, aukeratutako ikasketa algoritmoa aplikatzen zaio, kasu honetan gradiente estokastiko beherakorra, eta irteeratik sarrerara, geruzaz-geruza zuzendu egiten dira pisu sinaptiko guztiak.

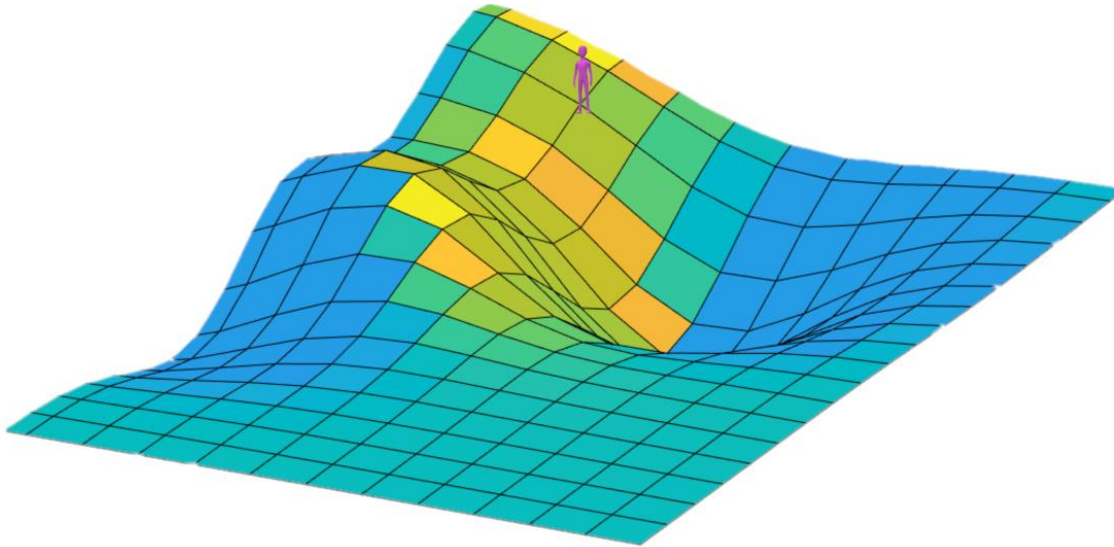


Irudia 4: Sare baten ikasketa prozesua[58].

Hurrengo puntuetan SGD algoritmoaren ezaugarri nabarmenenak komentatuko dira, era esplikatibo baten, matematikak alde batera utziz, algoritmoaren portaera orokorra azalduz, bilatzen den kontzeptua ulertu dadin. Gero IV atalean SGD brea atzean daukan matematika sakon azalduko da eta aurkitu diren fenomeno eta ahuldurak ere. Azalduko den lehenengo kontzeptua, SGD algoritmoaren portaera errearen aldean izango da, eta nola saiatu egiten duen sarearen pisu sinaptikoak zuzentzen. Bigarren puntuan, era klasikoan algoritmoak ematen dituen bi entrenamendu erak aurkezten dira beraien onura eta ahuldurekin.

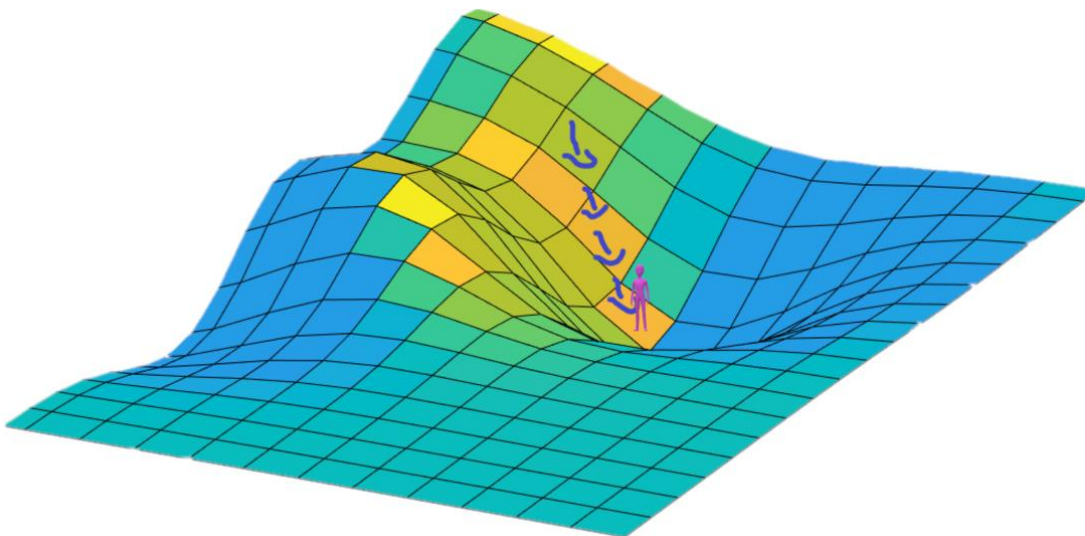
3.1.1. Mendi artean galdurik

Edozein sare neuronal bat harturik, sortu dezakeen errorea mendi artean dagoen haran baten forma hartu dezakeela. Errore altuenak mendi tontorrek izanik eta haran barrena errore minimoa. Sarea sortzen denean, normalean era aleatorio baten sortu egiten ditu bere geruzen pisuen baloreak, errore-haranean puntu ez jakin batean kokaturik[3]. Gainera haranaren forma ezagutzea ia ezinezkoa da, sarearen egitura berarekin lotuta dago eta sarea itsu dago daukan formarekiko. Ikasketa algoritmoa, haranean behera juteko bidea aurkitzeko gai den elementu bat izan behar du. Gradiente estokastiko beherakorrak haranaren malda sumatu egiten du, gradiente, errore funtzioaren lehengo deribatua izanik, honen malda adierazten duelako[3]. Gradiente hau jarraituz, harana pausoz pauso jaisten joango da norabide berdinean, gradiente aldaketa bat sumatu arte, non norabidea aldatuz aurrera jarraituko du haranaren puntu minimoa aurkitu arte[3]. Irudia 5-n ikusi daiteke azaldutakoaren irudikapen bat, non paisaia sarearen errorea bezala hartu daiteke eta pertsona, gradiente estokastikoaren isla den, bere hasierako errearen puntua, mapan kokatuta dagoen puntua izanik.



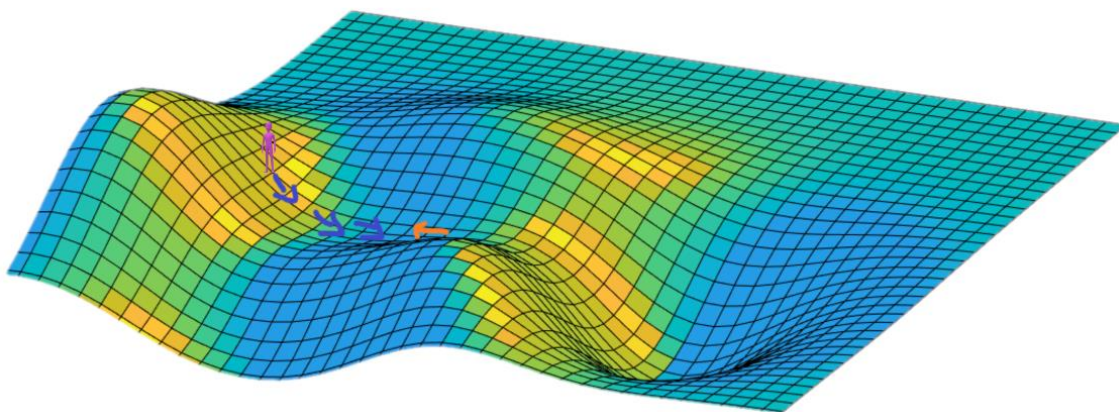
Irudia 5: Sare neuronalaren errorearen harana.

Algoritmoaren iterazio bakoitzean, sarearen errorearen kokapena haranean barrena sartuko da, minimo globala bilatuz. Emango duen pausoaren luzera, alpha deritzon parametro bati ezker erregula daiteke. Sortzen den pausoa oso handia bada, algoritmoa azkar jaitsiko du haranaren malda, baina behean egonda zail edukiko du minimo absolutua aurkitzeko hurbil dagoela, minimoa igaro egingo du baina ez da bertan gertuko, txikia bada, ostera, minimoko zonaldean egonda erraz aurkituko du, baina bertara heltzeko iterazioak asko igoko dira, informazio gehiago IV eta V ataletan. Irudia 6-n algoritmoa nola maldan behera joan den ikusi daiteke. Emandako urrats bakoitza gezi urdinekin irudikatu dira. Gezien norantza beti gradiente txikienera joera duen norabidean joango dira.



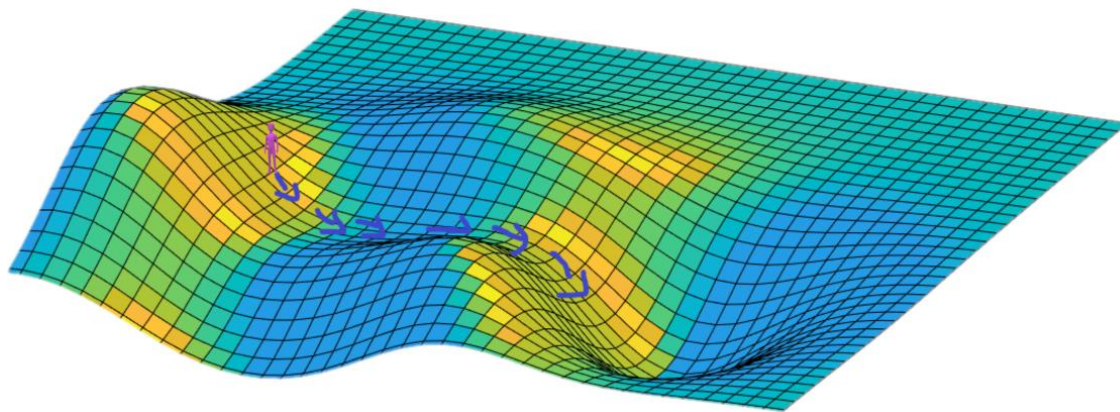
Irudia 6: Algoritmoa maldan behera bidaia egin du.

Posibilitate bat dago non sarearen hasierako errorearen puntutik abiatuta, haran txiki bat aurkitzea, minimo lokal bat, aurkitzea, eta sarea bertan geratzea gradientearen bitartez sumatzen den malda seinuz aldatzen duelako, atzerako mugimendu bat sortuz eta bertan trabatuta geratuz. Irudia 7-n ikusi daitekeen moduan, sarearen hasierako errorearen puntua minimo absolututik urrun kokatu da eta tarteko minimo lokal bat igaro beharko luke bertara iristeko, baina gezi laranja erakusten duen moduan, minimo lokala eta geroko pausua atzerantz emango du gradientearen balorea positibo bihurtzen hasiko delako eta algoritmoak, minimotik urruntzen ari dela sumatuko du, atzerako martxa emanez eta bertara bueltatuz.



Irudia 7: Algoritmoa trabatuta geratzen da minimo lokal batean.

Desiragarri ez den egoera hau ekiditeko, inertzia momentu bat gehitu egiten zaio, algoritmoari norantza berdinean jarraitzera derrigortuz, inertzia agortu arte. Era honetan, inertzia nahikoa pilatu bada, nahiz eta minimo lokal bat aurkitu aurrera jarraitu du, minimo absolutuaren bila. Alpharen kasuan bezala, inertiaren parametroa ondo doitzea oso garrantzitsua da, txikiegia bada, aurreko kasu berean egongo da algoritmoa, edozein minimo lokal baten oztopotuta geraturik. Oso handia bada, ostera, minimo absolutura iritsitakoan inertzia haundia izango du algoritmoa eta dagokion posiziotik aldentuko da, emaitza traketsak sortuz. Irudia 8-n ikusi daiteke nola inertiari ezker algoritmoak minimo lokala gainditzen duen eta minimo absoluturantz bidea hartzen duen.



Irudia 8: Algoritmoa, inertiari esker, minimo absolutura doa.

3.1.2. *Banan-banan edo multzoka*

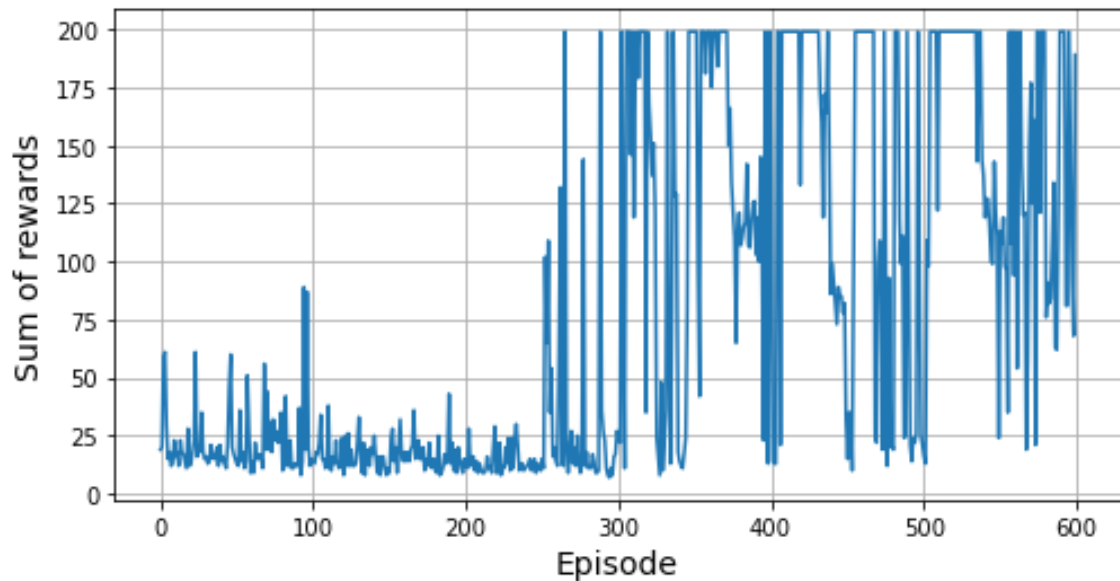
SGD algoritmoaren Stochastic hitza, entrenatzerako orduan, pasada bakoitzean aukeratzen den patroia era aleatorio baten sortzen delako dago aukeraturik[3]. Honek esan nahi du, entrenamendu epoka batean sarea entrenatzeko dauden patroia guztiak banan-banan pasako direla saretik, eta bakoitza sortutako errorearekin gradiente algoritmoa aplikatuz sarearen pisuak zuzenduko dira. Hurrengo puntuetan ikusiko den moduan, honen puntu ona, sarea era oso zehatzean ikasteko ahalmena du, baina denboran garesti atera daiteke eta gehiegi entrenatzeko aukera egon daiteke, begiratu 3.3 puntua.

Aukeran beste metodo bat dago, non patroiak tokatzen den epokarako era aleatorio baten ordenatu ostean, multzoka zatitu egiten da patroia kopurua eta gradiente algoritmoa bakarrik erabiltzen da multzo bateko elementu guztiak pasa ostean, hauen batezbesteko Loss funtzioa erabiliz erreferentzia moduan, sarearen pisu sinaptikoak zuzentzeko. Era honetan, ikasketa prozesuaren denbora pisua leundu egiten da, gradientearen bitartez pisu sinaptikoak zuzentzea operazio garesti bat delako denboran. Bestalde, gehiegizko entrenamendua ere murriztu egiten da, zuzenketa leunagoak aplikatzen baitira pisu sinaptikoetan. Alde txarrean, segun eta zer multzo tamaina aukeratzen den ikasketa prozesurako, sarea beharrezko diren modeloaren detaile batzuk ez ikasteko arriskuan dago, horrek eduki ditzakeen kontra-ondorioekin, ikusi 3.3 puntua.

3.2. Sorospen bidezko ikasketa eta algoritmo genetikoak

Puntu honetan tesi honen ikuspuntutik at geratu diren sare neuronalen ikasketa metodoak azalduko dira. Alde batetik, sorospen bitartez ikasten duten algoritmoak daude, non, "eragile" bezala ezagutzen den software programa bat, bere ingurumenari behaketa bat eginez, eragitea ezberdinak egiten ditu eta horien efektuagatik sorospen kantitate bat jasotzen du[72]. Sorospen honen bitartez eta jasotako behaketaren datuekin programa ikasi egiten du. Sistema "politika" bezala ezagutzen den estrategia erabiltzen du eta politika on bat lortzea da ikasketa metodo honen zuntzeko elementua[72]. Sare neuronaletara eramanda, politika sareak ematen duen erantzun estatistikoa bezala hartu daiteke. Loss funtzioa, sarea eragiketa batzuk egin ostean, sorospenaren bitartez lortu daiteke, honen gradientea kalkulaturik eta ikasketa ikuskatuaren kasuan bezala, edozein gradiente beherakorreko algoritmo bat erabili daiteke, ikasketa ikuskatuaren kasuan bezala, sarearen pisu sinaptikoak zuzentzeko. Ikasteko era honi Q-Learning deritzo[72].

Ikasketa metodo honen ahulezien artean, patroien irteera objektiboa ez edukirik, sareek epoka ugari behar dituzte errore txikietara heltzeko. Ere beste problema bat sortu daiteke, "ahazte katastrofiko"[72] bezala ezagutzen dena, non sorospen maximoetara heldu orduko, ikasketa hurrengo epokan, sorospen baxuetara berriro jaistea. 3.3-n ikusiko den moduan, sareak gehiegizko entrenamenduko puntura heltzen deneko antzerako egoera da, baina bapateko era baten gertaturik. Irudia 9-n ikusi daiteke sorospen bitartez Q-Learning entrenamendua erabilia, nola hasiera baten sorospen baxua lortzen duen, batzuetan sorospen altuagoa lortuz baina, bat-batean, berriz beherantz joanda. 250 epoka inguruan, sorospen maximora heltzeko ahalmena lortzen du baina epoka bat geroago berriz ere balore baxuetara bueltatu egiten da. Momentu horretatik aurrera, ikasketaren egonkortasuna galdu egiten du, noizbehinka berriro sorospen maximora heldu egiten da baina, epoka gutxi batzuk pasata, berriz hau galdu egite du.

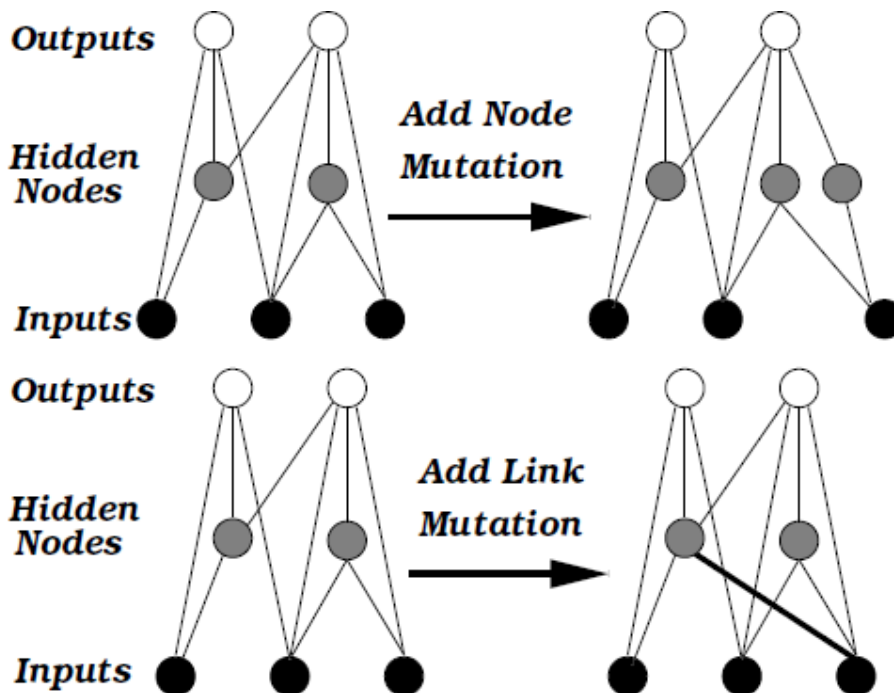


Irudia 9: Galera katastrofikoa erakusten duen algoritmo baten entrenamendua[72].

Bestalde, algoritmo genetiko moduan ezagutzen diren ikuskaketa gabeko algoritmoak dira. Kasu honetan, hasierako puntu bezala, kopuru definituko sare ezberdinak hartzen dira, denak topologia berdina edukirik edo era aleatorio baten erabakiak[7], [156]. Sare guztiak lanean jartzen dira eta entrenamendu epoka baten ostean, txapelketa algoritmo baten bitartez, hoberen lan egin duten multzoa aukeratu egiten da, beste guztiak deuseztatuz. Gelditu direnekin beraien artean gurutzaturik edota mutazioak aplikaturik, sare berriak sortzen dira hasierako multzoa bete arte. Prozesu hau, belaunaldi bezala ezagutzen dena, adostutako kantitate batean errepikatzen da, azkeneko belaunaldiko sarerik onena, sare definitibo bezala hautaturik[156]. Lan egiteko era hau naturan ikusitako eboluzio genetikokoaren antzekoa da, non hoberen egokitu diren norbanako bakoitza, ondorengoa izateko aukera eduki dezakeen, egokitu beharreko egoerara hobeto aurre egiteko aukera izango duena. NEAT[7], [157], [158] algoritmoa azkeneko multzo honetan sartu egiten da. Algoritmo honetan sarearen topologia eta barreneko pisu sinaptikoak zuzendu egiten ditu entrenamenduan zehar. Teknika hau arrakasta haundikoa izan da bideo-jolasen munduan[158]-[160], [160], [161].

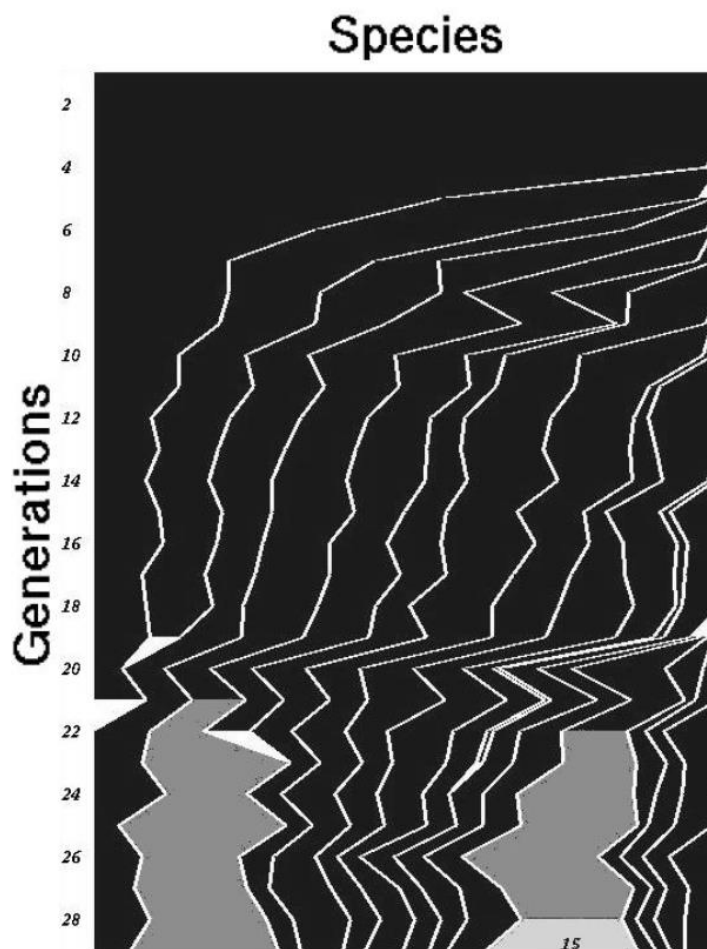
Esan den moduan NEAT algoritmoak sare minimo bat harturik honen pisu sinaptikoak eta tamaina doitu egiten ditu belaunaldiz-belaunaldi, ontzat eman daitezkeen emaitzak sorturik denbora bornatu baten. Bere definiziotik 2002-n, aldaketa eta optimizazio ugari izan ditu, aplikazio kasu bakoitzerako doitu[162]-[173]. NEAT, bere arbaso diren algoritmoek zeuzkaten problemak zuzentzeko sortu zen, adibidez, oso antzeko erantzuna eman duten bi gurasoen ondokoek sortu dezaketen informazio galera, edo, NEAT pisu sinaptiko aleatorioak dituen guttiz-konektatutako geruza bakarretik hasten dela eta ez pisu sinaptiko aleatoriodun guttiz-konektatutako geruza kopuru aleatorio batetatik[7], [80], [174]-[176].

Eboluzio prozesua bi mutazio prozesuetan datza[80], [175], [176], alde batetik, izkutaturiko geruzetan nodo berriak sortuz eta bestetik nodoen arteko konexio berriak sortuz. Guzti hau, eboluzio hasierako sare neuronal errazetik abiatuta, problema berdin bat ebazteko, pisu sinaptiko gutxiago erabili beharra ekartzen du normalean. Irudia 10-n ikusi daitezke bi mutazio era hauek.



Irudia 10: NEAT algoritmoko mutazio erak[175].

NEAT ere denbora marka bat txertatzen dio mutazio bitartez sortzen duen gene bakoitzari. Era honetan, bi banako gurutzatzean, beraien genek sortu izan zireneko denbora marka mantendu egiten dute. Honela, marka berdineko genek beraien artean lerrokatuko dira eta marka ezberdinak dituztenak guraso optimotik hartuko dira edo era aleatorio baten hautatuko dira[175]. Azkenik NEAT espezie ezberdinak sortzen ditu bere ebasketa populazioaren artean. Hau, berriro ere, geneei inposatutako denbora markari ezker sortu egiten da, espezie barruko izakien artean gurutzamenak eginez, era aleatorio baten egin beharrean populazio barruko izaki guztien artean era aleatorio baten aukeratu beharrean. Honek, ebazpenen egokitasuna, ektuktura berriak sortzerakoan, ez disolbatzea ekartzen du[175]. Irudia 11-n ikusi daiteke nola belaunaldiz-belaunaldi espezieak agertzeko joera azaltzen den, kolore argiagoak dituztenak ebazpen egokiagoak ematen dituztenak dira. Ikusi daitekeen moduan, belaunaldi berriek espezie berri gehiago sortzeko joera daukate, egokitasun gutxien daukaten espezie zaharrak desagertzeko joera daukatelarik. Kasu partikular honetan, 28-garren belaunaldian, 2 espezie egokitasun haundiagoa eskaintzen dute, besteekin konparaturik[157].



Irudia 11: Espezie ezberdinen bilakaera belaunaldiz, belaunaldi.[157].

Algoritmo genetikoetan, lehen azaldutako sorospen bidezko ikasketa erabili egiten da, non lortutako sorospena izango da erabiliko dena belaunaldi horretako baheketa egiteko, non sorospen gutxien sortu duten izakiak deuseztatu egiten diren, sorospen haundia lortu dutenak bakarrik mantenduz[175], [177], [178]. Sistema egokitasuna definitzen duen funtzioa, "Fitness function" ingelesez[179], garrantzitsua da oso sareak sortuko duen modeloarentzako. Kontutan hartu behar da ere, ahalik eta egokitasun funtzio honi ahalik eta dei gutxienekin modelo optimo bat lortu behar dela, normalean bere exekuzio denbora altua izateko joera baitu. Sortutako sare neuronal berri bakoitza, entrenatu beharra dago gero aztertuta izateko eta hau belaunaldi bakoitzeko izaki bakoitzarekin egin beharreko lan bat da, denbora asko eska dezakeena[179]. Hurrengo belaunaldia sortuko duten hautagaiak, lehen esan den moduan, sorospen haundiena lortu duten izakien atean aukeratuko dira edo sorospen positiboa sortu duten izakien artean era aleatorio baten aukeratura[179].

Honetaz aparte, algoritmo hauetan, ere beharrezkoa da gelditze estrategia bat definitzea, hau da, noiz eboluzioa geratu behar den eta ez da sare belaunaldi berri bat sortuko. Ikuskatutako sareetan bezala, hau balore finko bat izan daiteke, atzera hedatzearen epoka kopurua bezala, hemen belaunaldi deituta, edo egokitasun funtzioan minimo batera heldu izana, edota funtzio horretan asetahun limite batera heldu izana, hau da, aurreko belaunaldiarekin alderatuta, lortutako egokitasun mahaiaren diferentzia atalase batetik behera egotea, emaitza hobereana edukitzeko probabilitate altua edukiz[179]. Eboluzio prozesua bukatu ostean, lortutako emaitzaren egokitasuna mundu errealean egiaztatu beharko da. Ikuskatutako ikasketan bezala, beharrezkoa da sareak inoiz ikusi ez dituen patroiak erabiltzea, sarearen balioztapena egiteko[179].

Algoritmo ebolutiboek, hasierako parametro sorta bat definitu behar dituzte, eboluzioan zehar izango duen portaera definitzeko, sareko elementuen mutazio ratioa bezala[156], [179]. Parametro hauen eskuz egindako ezarpena ez da batere erraza, parametro estatikoek, sarea ez hain optimoa izatera eramanez dezakeelako edota optimizazio hori lortu ahal izateko denbora gehiegi inbertitu behar izatea.

Hau ekiditeko, badaude aplikatu daitezkeen teknikak daude. Haien artean, sarearen eboluzioan era positiboan eragiten duten parametroei erregela beherakor bat aplikatzea, lineal edo esponenziala, hasierako influentzia haundia izan dadin, baina belaunaldiak joan ahala geroz eta txikiago bihurtu, eboluzioaren pausuak murriztuz eta era kontrolatuago batean eman daitezen, optimotik gertu dagoenean[179]. Posible da ere, erregela konplexuagoak aplikatzea Rechenberg erregela bezala[179], parametroak aldatzeko aukera ematen duena egokitasun funtzioaren emaitzen arabera.

3.3. Sare neuronalek gainditu beharreko oztopoak

Sare neuronalek oztopo ezberdinak gainditu behar dituzte bere ikasketa eta lan egiteko orduan. Hauen erabilera normalean, lortu beharreko modelo matematiko edo fisikoa argia ez denean erabiltzen dira[58], modelo bat eginez indukzio bitartez ikasiz emandako patroientatik. Ikasketa honetan badaude hasieratik agertzen diren problema batzuk:

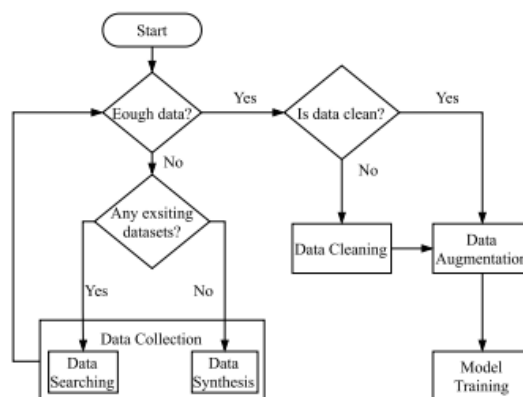
→ Ikasketa patroiak ez daude loturik, gero sareak landuko dituen datuekin. Posible da erabiltzen diren patroien kopurua nahikoa ez izatea edota aniztasun nahikoa ez izatea benetan inferitu behar duen modeloa ondo irudikatzeko. Normalean, datu gabezia konpentsatu ahal izateko, patroia artifizialak sortzeko teknikak erabiltzen dira, hauen ezaugarri batzuk aldatuz, patroia berriak sortuz. Irudia 12-n ikusi daiteke normalean sare neuronal bat entrenatzeko erabiltzen den prozedura. Prozedura hau, hiru multzo haunditan banandu daiteke[180]:

- Datuen bilaketa, hau dagoeneko existitzen diren datu baseetatik datuak hartzea izan daiteke, edo, simulazio baten bitartez, datuak sortu.
- Datuen garbiketa: Datuak normalean, nahi ez diren zaratekin etortzen dira, diseinatzailearen lana da datu hauek garbitzea patroia moduan erabili ahal izateko.
- Datuen handiagotzea: Lehen esan den moduan, teknika ezberdinak daude patroia bihurtuko diren datuei, modifikapen ezberdinak aplikatu ahal zaizkie patroia berriak sortzeko eta esangura nahiko eduki dezaketenak, sarea ondo entrenatu eta gerora baliagarri izateko aukeratu zaion egorarako.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

- Sarea gehiegizko ikasketako estatura iristea. Sare bat datu gutxiegiarekin entrenatzen bada edo entrenamendu epoka gehiegi pilatzen baditu, posible da sareak lan datuak era orokor batean lantzeko ahalmena galtzea, ikasketa patroien ezaugarri oso espezifikoa ikasten ditu, gerora, datu errealekin lan egiteko orduan, hauek gaizki tratatzeko arriskua sorturik. Era klasiko baten, ikasketarako erabiliko diren patroien talde bat gorde egingo da, ikasketarako erabiliko patroiekiko esanguratsuak direnak. Era honetan, sareak ikasketa epoka bat burututakoan, gorde diren patroiek sarearen ikasketaren balidaziorako erabiliko dira, hauen errorea kalkulatu. Ikasketaren prozesuan balidazio patroien errorea joera gorakoa badu, ikasketa errorea beherakorra duenean, ikasketa prozesua geldiaraziko da, sarea gehiegizko ikasketaren sintomak islatzen hasiko delako.
- Sareak erabiltzen duen ikasketa algoritmoa ez bada egokia ikasi beharko duen patroientzako. Posible da, erabiliko den ikasketa algoritmoa nahikoa ez izatea ikasi behar diren patroien ezaugarriak ondo islatzeko sare neural barruan.
- Sare neuronalaren egitura egokia ez izatea planteatzen lan eremuarentzako. Sareak behar adina geruza ez izatea edota geruza bakoitzak dituen pisuen kopurua nahikoa ez izatea posible da. Hau gertatzerakoan, nahiz eta patroien kopurura handitu edota ikasketa epokak handitu, sarea bere topera heldu da ta ez da posible modeloaren hobekuntza bat, gehiegizko entrenamendura heldu gabe.

Normalean, oztopo hauetaz, sarearen diseinatzailea, sarea entrenatu eta gero aurkitzen ditu, horregatik garrantzitsua da oso aldez aurretik aurkitu daitezkeen oztopoak identifikatu eta konpontzea.



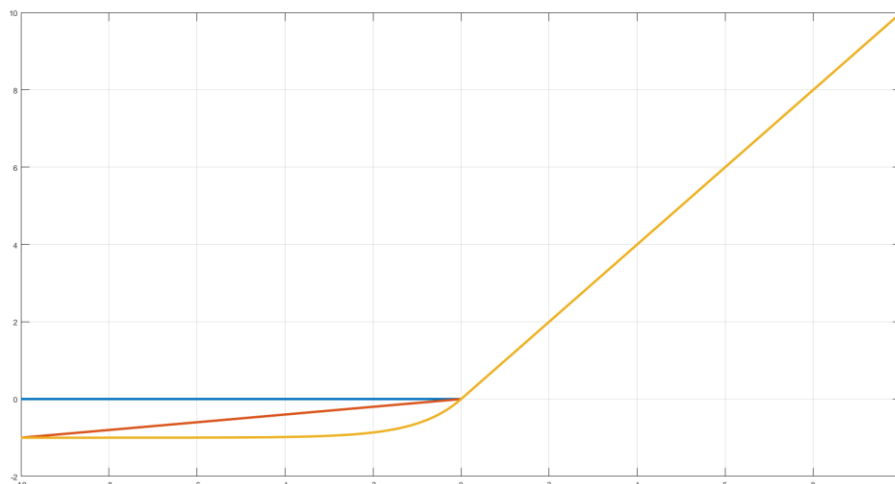
Irudia 12: Sare neuronal bat entrenatzeko beharrezko diren patroien bilaketaren eskema[180].

3.4. Sare neuronalen optimizaketak

Urteetan zehar, Machine Learning eta sare neuronalak aztertu ahala, zenbait aldaketa proposatu egin dira beraien bilakaera geroz eta hobegoa izateko. Hobekuntzak sareen elementu ezberdinetan fokatu egin dira, baina beti sareen entrenamenduan ikusiriko ahuldurak zuzendu nahian. Era orokor batean, gradientearen desagerpena, ikasketan zehar Loss funtzioaren minimo lokaletan ez geratzea edota sareak informazio gehiago gordetzeko ahalmena edukitzea, modelo osoagoak sortuz eta gehiegizko ikasketan erori gabe. Era orokorrean, hobekuntza hauek aplikatzea, sarearen konputazio kostua, bai entrenamenduan zehar, bai produkzio lanean egonda, igo egiten du. Eragin hau, segun eta sarea lan egingo duen sistema zein den hausnartu behar da, jasanezina bihurtu ahal delako, bai entrenatzerako orduan edo garrantzitsuago, lan egiteko momentuan. Hurrengoko puntuetan ezagun bihurtu diren optimizaketak agertzen dira.

3.4.1. ReLu-ren aldaketak

ReLU algoritmoa, entrenamenduan zehar, duen sarreraren parte negatiboa ekiditeagatik, bere menpe dauden neuronen kopuru bat "hiltzera" eraman ditzake. Honek esan nahi du momentu bat iritsi daiteke non neurona horiek eskaintzen duten balorea beti 0 izango dela, bere pisu sinaptikoaren eta offsetaren balioak beti erantzun negatibo bat sortuko dutelako eta ReLu-ak beti 0 bat aktibatuko duelako. Gero, gradienteagatik geruza zuzentzerakoan, Ora geratu diren pisu sinaptikoak ez dira zuzentzen, betirako baliogabe geraturik[72]. Horregatik autore ezberdinek leakyReLU bezala ezagutzen diren algoritmoak planteatu dituzte[72], [181]. Aldaketa hauetan, sarrera negatiboetarako, "galera" bat proposatzen da, irteeran balore bat islatuz eta gero gradientearen medio, zuzenketa bat jasotzeko aukera izanik. Hauen antzekoa, ELU deritzon algoritmoa da[72], [182]. Kasu honetan, parte negatiboko funtzioa -1 saturatzen den funtzio esponentzial batez osaturik dago. Sortzen diren emaitzak oso onak bilakatzen dira, konputazio astunagoa bilakatzen dela kontutan harturik. Irudia 13-n ikusi daiteke hiru funtzioen forma, nola ReLuak edozein balore negatibo mozten duen, nola leakyReLU era lineal baten jasotzen dituen balore negatibo horiek eta azkenik ELU, -1 asetzen duen funtzioa izanik.

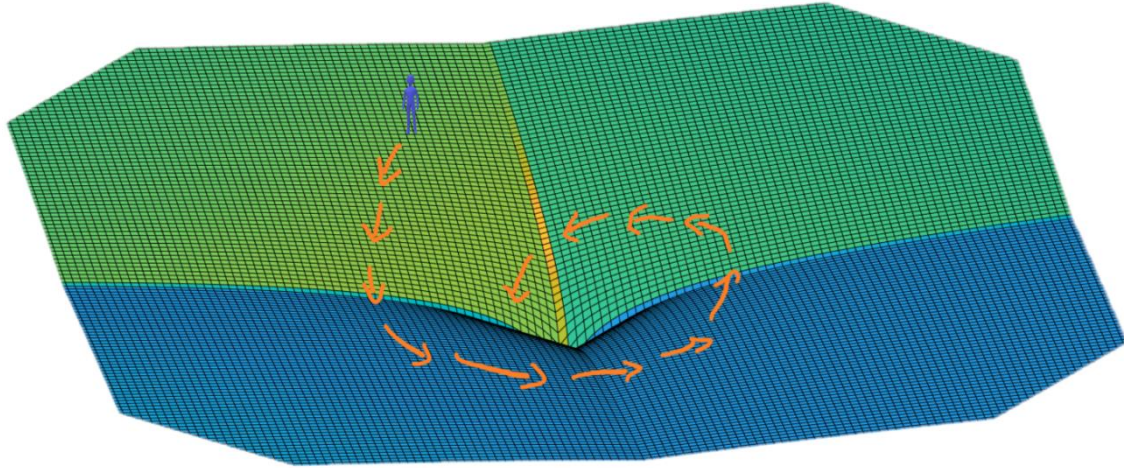


Irudia 13: RELU(urdinez), leakyRELU(gorritz) eta ELU(laranjaz) funtzioen forma.

Azkenik, 2017-n Günter Klambauer et al[72], [183] SELU, Scalated Exponential Linear Unit, proposatu zuten. Aktibazio funtzio berri honekin, sarearen geruza sakonetan erabilirik sarea bere kabuz normalizatzeko joera izango du, prozedura geruza bakoitzaren irteera 0ko batezbesteko balioa izango du eta 1ko desbideratze estandarrekin[72]. Hau lortu ahal izateko, sarea entrenatu baino lehen ezaugarri batzuk bete behar ditu. Alde batetik, sare sekuentzial bat izan behar du, ezin du geruza paralelorik eduki behar ezta aurrera edo atzera saltorik ere. Bestetik, erabilitako geruzak guzti konektaturikoak izan behar dute eta LeCun normalizazioaz abiarazita egon behar dute[72].

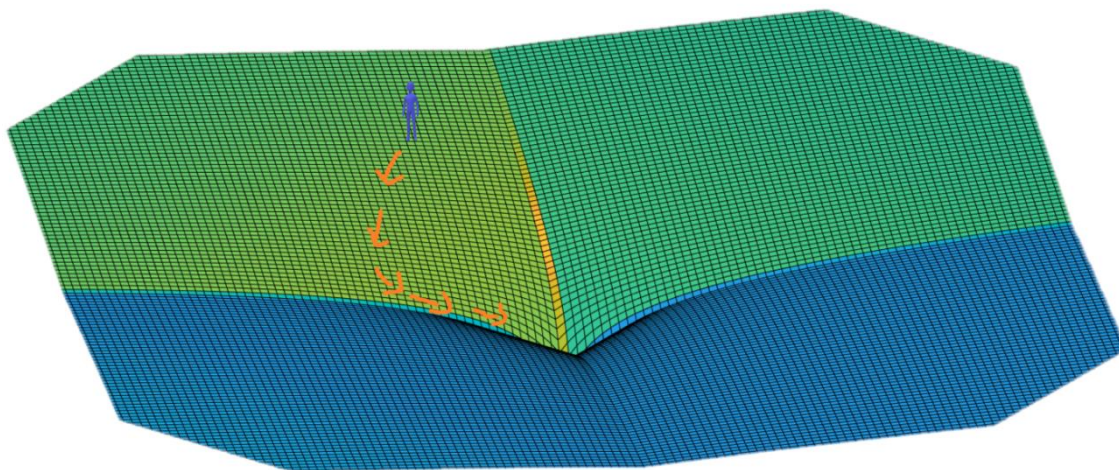
3.4.2. AdaGrad, RMSProp eta Adam

SGD algoritmoaren erabileran, Loss funtzioaren minimo globala bilatzeko bidaian, posible da funtzioa minimo lokal batean geratzea oso posible da. Hau ekiditeko, SGD algoritmo berari momentu bat aplikatzen zaio, algoritmoari inertzia gehituz, eta minimo lokal hauek gainditzen laguntzen duen bitartean. Momentu honen arazoa da, Loss minimoko zonalde lauan dagoelarik, betiko errebote eginez bere inguruan, baina inoiz honetara iritsiz. Efektu hau ekiditeko, algoritmo ezberdinak garatu egin dira, algoritmoa zonalde lauan sartu egiten den momentuan, gradientearen momentua gutxiagotuz edo honen norantza aldatuz. Irudia 14-n ikusi daiteke inertzia momentu normal bat aplikaturik, zonalde lau baten egonda, minimo absolutu batera juteko edukiko lukeen ibilbidea. Inertzia momentua nahiko handia bada, sarearen errorea orbita moduko bat sortzeko arriskua dauka, minimo absolutua inoiz ukitu gabe.



Irudia 14: Momentu normalarekin egingo lukeen bidea.

Hauetan garrantzitsuenak AdaGrad[72], [184] familiako algoritmoak dira. Algoritmo hau, gradientearen bektorea txikiagotzen du, errore altuko zonaldeetan, eta mantendu egiten du lauagoak direnetan. Honi ikasketa tasa adaptatiboa deritzo. Alde txarrean, AdaGrad algoritmoa, segun eta zer kasu entrenatzen ari den, hau goizegi gelditzeko joera izan dezake. Joera hau zuzentzeko RMSProp[72] garatu zen, azkenengo gradienteaz bakarrik elikatuz, hasierako gradiente guztiak deuseztatuz denbora bat pasata. Azkenik, Adam deituriko algoritmoa sortu zen[69], [78], [79], [185]–[195]. Bere izena, momentu balioztapen adaptatibotik dator eta SGDren momentu klasikoa eta RMSProp bateratzen ditu. Azkenengo hau, oso era errazean erabiltzeko ahalmena dauka, bere hiper-parametroen ezarpenak ez du zertan oso fina izan behar. Beti ere, kontutan hartu behar da, algoritmo hauek, bilakaera astunagoa dutela sarearen prozeduran SGD algoritmoa baino. Irudia 15-n ikusi daiteke algoritmo hauekin lortu nahi den errorearen ibilbidea, inertiagatik espiral moduko ibilbidea sortu beharrean 90° tik gertuago dagoen ibilbide bat sortu nahi da minimo absoluturantz.



Irudia 15: AdaGrad, RMSProp edo Adam-ekin lortu nahi den bidea.

3.4.3. Erregularizazioak

Sarearen ikasketan zehar, badaude egin daitezkeen erregularizazio metodo batzuk, sarearen gain entrenamendua gutxiagotu dezaketenak. Erregularizazio hauen artean erabilitakoena Dropout deritzona da, honetaz, IV eta V puntuetan gehiago hitz egingo da. Aplikatu daitezken beste erregularizazio batzuk, L1 eta L2 bezala ezagutzen direnak dira.

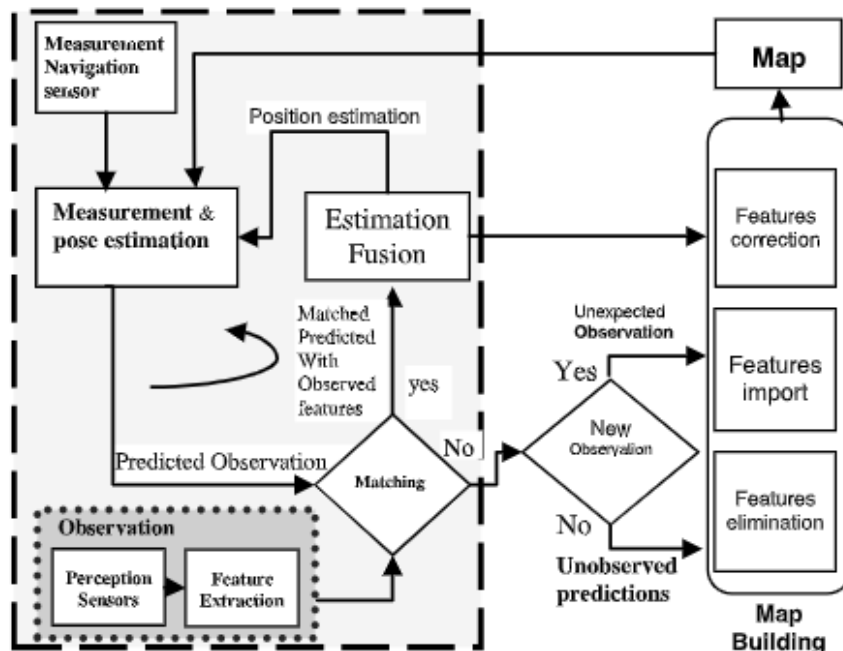
L1 norma erabiltzen duen erregularizazioari Lasso (Murrizketa absolutu txikienaren eta aukeraketaren operadorea) bezala ezagutzen zaio[196]-[220]. Erregresio linealaren edo SGD baten gainean aplikatzen den faktore gehigarri bat da non bilatzen dena modelo hedatu bat sortzea da, hau da, pisu sinaptiko ugari 0 balorea edukitzea eta era sakabanatu baten, balore printzipalak edukitzea. L2 norma erabiltzen duen erregulazioa Ridge[221]-[245] edo ertzekoa deritzo. Aurreko kasuan bezala Erregresio lineal edo SGD baten gainean aplikatutako termino gehigarri moduan aplikatzen da. Aurreko kasuan bezala, ikasketaren prozesua hobeto eta irteera orokorragoak sortzeko balio du.

3.4.4. Batch Normalization

Batch Normalization[72], [246], Sare neuronal sakonetan aplikatzen den metodo bat da. Metodo hau ezkututako kapa bakoitzaren sarreran eta irteeran aplikatzen da. Algoritmo honek, sarreraren sorta bat harturik, hauek 0-n erdiratzen ditu eta normalizatzen ditu. Ekintza hauek eginda, sare neuronalaren portaera hobetzen da, gradientearen desagerpena murriztuz, eta hasierako pisu sinaptikoen eta aktibazio funtzioaren independenteago eginez. Alde txarrean konplexutasun gehiago gehitzen dio sareari eta prozedura denbora era nabarmen batean igotzen du.

3.5. Sare neuronalen aplikazio adibidea: Nabigazio autonomoa

Ibilgailu baten nabigazio autonomo zuzen batek, honen kokapena eta ingurumenaren ulermena, ezinbestekoak ditu. Hau lortu ahal izateko, SLAM, Simultaneous Location and Mapping, metodoa aplikatu dezake. Metodo honekin ibilgailuak ez du zertan dagoen zonaldeko mapa aurretik kargaturik izan behar edo hasiera batetik bere posizio zehatza jakin behar, GPSkin lan egiten duten kanpoko aplikazioak ez bezala, bazik eta nabigazio sentsoreen bitartez jasotzen duen informazioa landuz, mapa bat eraikitzeke eta bertan non kokatuta dagoen jakiteko ahalmena edukitzea, gizakiaren laguntzarik gabe. Teknika hau, 20 urte baino gehiagoz erabilia izan da eta ingurumen ezberdinetara egokitu da, bai kanpoko ingurumenetan, barrukoetan edo ur azpikoetan ere[247]. Irudia 16-n ikusi daiteke SLAM algoritmo baten funtzionamenduaren eskema.

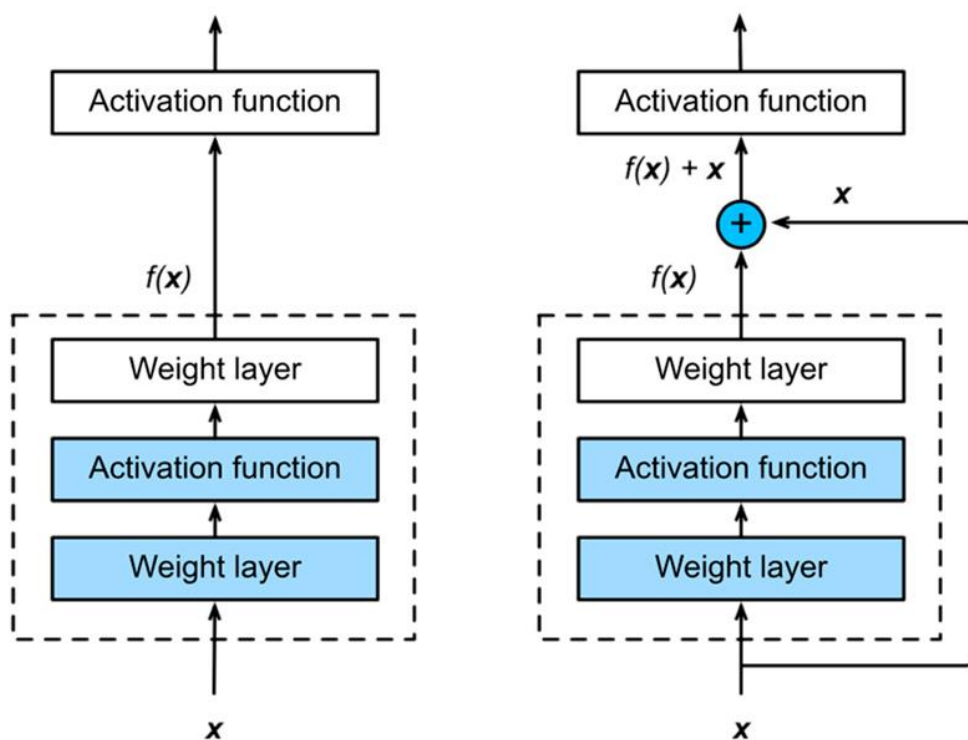


Irudia 16: SLAM-en lan egiteko eraren eskema[247].

SLAM erabiltzen duten aplikazio gehienek Bayesen filtroak erabiltzen dituzte ebazpenerako, zatiki filtroak, Particle Filters, edo hedatutako Kalman filtroak, Extended Kalman Filter, bezala. Filtro hauekin, mapak eratu egiten dira eta gero ibilgailuak dituen sentsoreen bitartez[247]-[251]. Badaude beste aplikazio batzuk, ingurumen ez-Gaussianoetan eta ez-linealetan lan egiten dutenak[248]. Badaude beste lan batzuk kameratan oinarritzen direnak, beste sentsore batzuen gainera, batez ere beste sentsoreek izan ditzaketen joera aldaketengatik, adibidez ingurumeneko argitasunaren aldaketekin[249]. Metodo honekin eman daitezkeen ikuspuntu kantitatea haundia da, Huenag eta Dissanayakes beraien editorialean dokumentatzen duten bezala[250].

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

Nabiagazioaren abidibe bezala Teso-Fz-Betoño et al. Proposatzen duten ingurumen industrialeko barrutietan lan egiteko AMR, Autonomous Mobile Robot, ibilgailua dago[94]. Kasu honetan, ReNet-18 motako sare neuronal konboluzional bat erabilia eta sarrera sentso moduan web kamera bat erabilia. ResNet-18, izena hartu egiten du bere geruza konboluzionalen artean aurreko geruzako sarreraren parte bat hurrengo geruzako sarrera bezala pasatzen zaiolako, hondar bezala ezagutzen delarik, eta, era honetan, gradientearen desagerpena murriztu egiten da. Irudia 17-n ikusi daiteke sare normal baten eta hondarra erabiltzen duen sare baten artean.



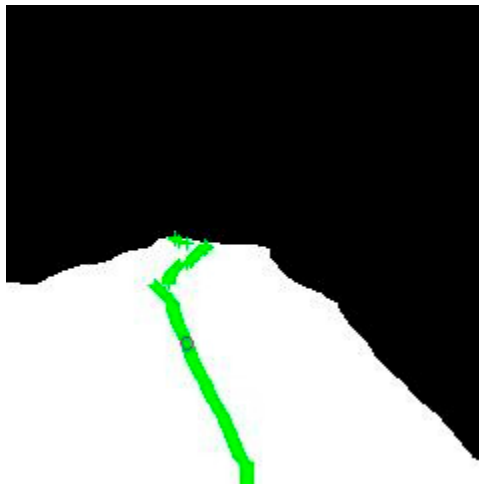
Irudia 17: Hondar sarea, sare normalarekin alderatuta[94]

Kameratik hartzen diren irudiak, segmentazio semantiko bitartez sailkatu egiten dira, hau da, irudiaren pixel bakoitzari esanahi bat ematen zaio, hau da, irudiaren parte bakoitza, zorua den edo pareta/oztopo bat den. Horretarako entrenatzeko erabiltzen den irudi patroia bakoitzean pixel bakoitzari zer motakoa den definitu behar zaio. Irudia 18-n ikusi daiteke honen adibide bat.



Irudia 18: Adibide irudi bat non urdinez zoru bezala klasifikatu diren pixelak daude eta laranja/gorri pareta[94].

Azkenik, informazio hau edukita, sistemak bere bidea egiteko funtzioa definitu behar da, kasu honetan, ResNet-18 sareak kamerak atera duen azkeneko argazkia segmentatu ostean, argazkiaren lerro bakoitzeko zenbat zoru dagoen kalkulatu egiten da eta bere batezbesteko puntutik igarotzera eramaten du robota[94], [252]. Edozein lerrotan bi zoru zonalde sortzen badira oztopo bat dagoelako zonalde haundienetik pasatzera eramango du. Irudia 19-n ikusi daiteke aurreko iruditik ateratzen den bidea erabakitze funtzioaren irteera[94], [252].



Irudia 19: Irudia 18-tik ateratzen den bidea egiteko funtzioa[94].

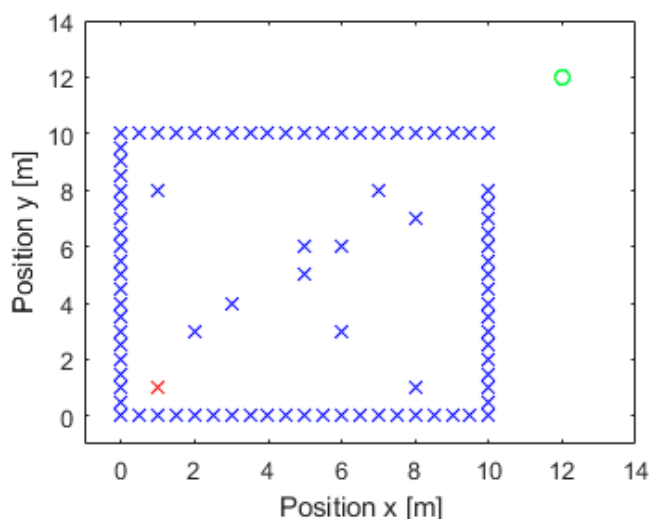
Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

Bide berdinetik jarraituta, Cabezas-Olivenza et al.[93] beste AGV, Automated Guided Vehicle, ibilgailu bat planteatzen du, non kamera aurrera begiratu beharrean zorura begira jarri egiten da eta koloredun zinta bat erabiliz AGVak jarraitu behar duen bidea markatzen zaio. Metodo honekin, sare neuronal konbolutibo baten bitartez AGVak eraman behar duen ibilbidea balioetsi egiten da eta Stanley algoritmoaren bitartez [93], [253], hegal eta luzetarako kontrola aplikatzen da, ibilbide hori jarraitzeko. Azkenik, Lyapunov metodoaren bitartez algoritmoaren estabilitatea eta egonkortasuna probatzeko. Irudia 20-n ikusi daiteke sortutako ibilbidea zintaren gainean. Ondorio bezala, algoritmo egonkor bat sortzeko ahalmena dagoela hartu egiten da.



Irudia 20: Zinta markatzailearen gainean kalkulatutako ibilbidea[93].

Hurrengo lanetan, sare konbolutibo baten partez, sorospen bitartez entrenatutako agente bat erabilia eta guztiz konektaturiko geruzez osatutako sareak erabilia, ibilbideak sortzeko ahalmena ere Lyapunov metodo erabilia aztertu egiten du[60]. Agenteak, DDPG, Deep Deterministic Policy Gradient, teknikarekin entrenatu egiten dira. Kasu honetan, oztopoz betetako gela batetik ateratzeko algoritmoaren ahalmenean jartzen da ikuspuntua. Irudia 21-n ikusi daiteke proposatutako mapa ikasteko, urdinez oztopoak adierazita, berdez iritsi beharreko puntua eta gorritz ibilbidea hasteko jatorrizko puntua.



Irudia 21: Proposatutako nabigazio mapa[60]

II. Teknikaren egoera

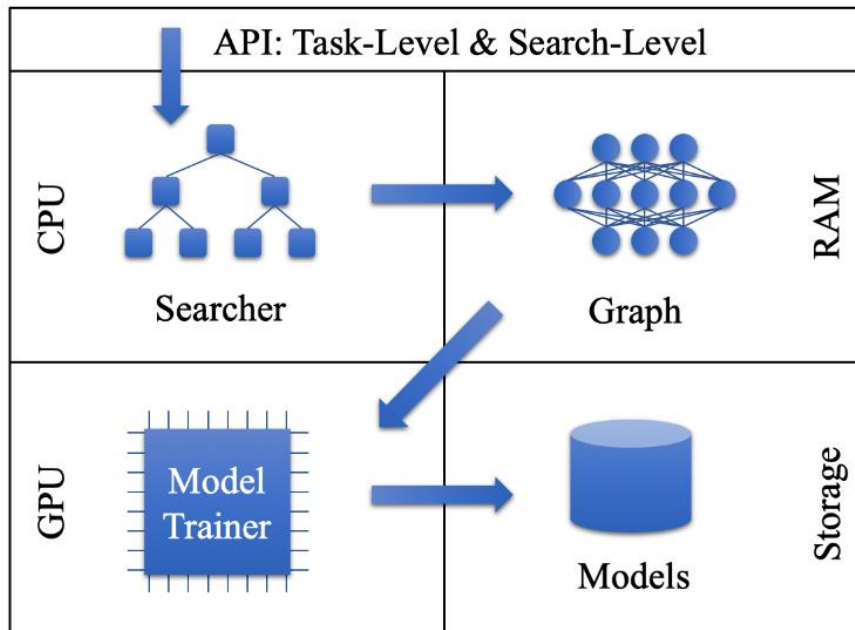
1. Sare neuronalen optimizaketa

Sare neuronalen optimizazio automatikoa (egitura, entrenamendua, eragiketa denborak,...) landu behar den atal bat da, sareak emaitza onak ematea lortu nahi bada denbora finitu baten[254]. Eskuz eraiketako metodo tradizionalak, saiakera-errore bitartez lortu egiten dira sareak lortu nahi deneko portaera lortu arte[180]. Normalean, lortzen diren sareak errekurtsio gehiegi eskatzen diote edukiera-mugatutako sistemei, hau ikusi daiteke Teso-Fz-Betoño et al[255] lanean, non jarraitutako sare arkitekturaren bilaketak 100 egun dirauten du, emaitzarik finenak lortu barik, eta ondorio moduan arkitektura honen bilaketa sareak entrenatu gabe erabiltzen dituzten metodoak erabili behar direla ezarri egiten da. X.He et al.[180] AutoML deritzoten sistema bat proposatzen dute, hurrengo puntuekin:

- **Datuen prestaketa.** Ikasketako datuen bilketaren inguruan gertatzen den dena kontutan hartzen da, datu birtualen bilketa ere sarturik. Datu birtual hauek segurtasunagatik datu errealak lortzea ezinezkoa denean ematen da, gidaketa autonomoaren kasuan bezala[180]. Datu prestaketan ere kontutan hartu behar da, entrenamendu datuen garbiketa, sarea kitzikatu dezaketen zaratak kendurik, eta datuen handitzea ere, non dauden datuetatik beste berri batzuk lortuz, beraien ezaugarriren baten oinarriturik, sarearen gehiegizko ikasketa ekiditeko[4], [180], [256], [257].
- **Ezaugarrien erauzketa.** Ikasteko datuak sareak ikasi behar dituen ezaugarriekin datoz. Batzuetan, ezaugarriak sortu beharko dira datu garbietan, besteetan, ezaugarri multzo bat, ezaugarri orokorrago bakar batean bildu beharko dira[180], [256], [257]. Sare neuronal sakonen betebeharrak bat, gordin dauden datuetatik ezaugarriak ateratzea da.
- **Modeloaren azterketa.** Azterketa hau normalean sarea entrenatu ostean egiten da. Normalean balioztatze datu multzo batez egiten da azterketa hau. Dagoeneko, beste metodo batzuk eraiki dira lan honetarako, kostu txikiagoa suposatzen dutenak, baina sarearen fidagarritasunean inpaktua daukate[180].

Sarearen arkitekturaren hobekuntzak bilatzeko bidean, autore ezberdinek proposamenak egin dituzte. Mellor et al.[258] proposatzen dute entrenatu gabeko sare baten irteerak ikertzea, ateratako irteeren eta aplikatutako sarreren arteko korrelazioa aztertuz. Geroz eta korrelazio altuagoa eduki, txartoago lan egingo du sareak, sarrera ezberdinek oso berdintsu diren irteerak sortzen dituztelako. Metodo hau aplikaturik, arkitektura ezberdinen artean hautatzea posible da konputazio esfortzu gutxiokin, txartoen portatzen diren arkitekturak deuseztatuz eta posible diren sare multzo txikiago batekin geratuz. Ala ere, geratzen diren sareak, entrenatu beharko dira eta beraien emaitzak konparatu, haien artean hautatzeko. Han Cai et al.[259] ProxylessNAS deitzen duten prozedura bat proposatzen dute, non, beraien hitzetan, konboluziodun sare neuronal bat entrenamendu denboran araztu dezakeena, entrenamenduaren errekurtsio konsumoan inpaktu txikia izanik. Beraien lanean, CIFAR-10 eta ImageNet liburutegiekin emaitza onak atera dituztela diote.

Ildo berdinean, Haifeng et al.[260], Auto-Keras deitutako Python eta TensorFlow-en garatutako framework bat proposatzen dute. Framework honek, optimizaketa Bayesino baten bitartez sare neuronalerako bilaketa eremu baten barruan, ezpen egokiena bilatzea. Beraien algoritmoa, sare neuronalez osatutako kernel baten oinarritzen da bere eraiketarako eta zuhaitz formako funtzio baten datuak eskuratzeko. Irudia 22-n ikusi daiteke proposatzen duten sareen bilaketa eskema, erabiltzailea, API-aren bitartez, prozedura hasten du. Bilatzailea, CPUa erabiltzailea sare baten proposamena eraikitzen du. Grafikatzailea, bilatzaileak sorturiko arkitekturarekin sare neuronal berri bat sortu egiten du RAM memorian gorderik. Gero, modeloa GPU-aren bitartez entrenatu egiten da eta hau bakatzen denean sortutako modeloa disko gogorrean gorde egiten da. Gordetako informazioarekin, bilatzailea eguneratu egiten da eta prozedura berriz hasi egiten da, bilaketa denbora bukatu arte, non. Proposatutako sare entrenatua eskuratuko den[260].



Irudia 22: Auto-Keras-en prozedura grafikoa[260].

Mingwei Zhang et al.[261] NAS-HRIS deitutako sistema bat sortu dute segmentazio semantikorako erabiltzen diren sareetarako. Azikliko den grafo baten oinarrituta, gradiente beherakor optimizatu bat erabiltzen dute eremu barruko bilaketa funtzio bezala eta hau NAS sistema irakasteko erabiltzea proposatzen dute. Azkenik, Teso-Fz-Betoño et al.[255], sare neuronalen bilaketa automatiko bat proposatzen dute era iteratibo batean, bilaketa Bayesinoa eta algoritmo genetikoak alde batera utzita, LiDAR bidezko nabigazio sistema baten kokapena estimatzeko, eta, ahal bada, kokapenerako erabili diren algoritmo klasikoak sare neuronalez ordezkatzeko. Sare hauek, guztiz konektaturiko motakoak dira eta bakoitza puntu kardinal bakoitzaren gehi errotazio puntua estimatzeko erabiliko dira. Ondorio bezala, sare neuronalen bilaketa astuna sorta daiteke, aldagai diren parametroen kantitatea altua bait delako, ala ere, sare neuronalen erabilera, LiDAR sistemetan erabilitako algoritmo klasikoak ordezkatzeko hautagai onak direla antzeman da eta zehaztasun haundiko ingurumenetan hauen arkitektura ebazteko NAS metodoak ontza ematen dira.

2. Sare neuronalen ulergarritasuna

Sare neuronalen ulergarritasuna, Machine Learning-en ikuspuntutik, sareak ematen dituen iragarpenak, gizakiek ulertu dezaketen era baten aurkezteko eta esplikatzeko gaitasuna da[84], [262]–[266]. Hartzen diren erabakien interpretatzea ikertze arlo korapilatsua da berez, sareak daukan konplexutasuna dela eta. Konplexutasun honi ezker sarearen emaitzak sortzen dira, baina honek zaildu egiten du irteeraren, sarreraren eta entrenamendu datuen loturak sumatzea. Ikasketa automatikoko algoritmo bat auditatu edo hobetzeko, beharrezkoa da hau ulergarria izatea, batez ere pertsonen edo makinaren segurtasuna jokoan badago[5], [55], [84], [267]–[281]. Ulergarritasun teknikak eta metodologiak, sare batek zergatik ez duen iragarpen bat ondo egin ikertzen laguntzen dute eta errore hauek zuzendu ere[263]. Badira kasuak non ez hain beharrezkoa ulergarritasun hau, inpaktu haundirik ez duten kasuetan edo oso ezagunak diren kasuetan, non bere datuekin irakatsitako sare baten irteerak oso arraroa den portaera bat izatea probabilitate oso baxua dela. Bestalde, sare baten portaeran aldaketa bat sortu nahiko lukeen edonor, sare horren emaitzak ulertzeko ahalmena izan beharko luke[55], [263], [265], [282], [283].

Sare baten ulergarritasuna metodo ezberdinengatik egin daiteke. Metodo batzuk, sarearen konplexutasunari mugak jartzen dizkiote, beste batzuk, modeloaren portaera ikertzen dute entrenatuta izan ondoren. Lehenengo metodoen multzoa, modeloari lotuta bukatzen dute eta bigarrenengoek, nahiz eta modelotik independenteagoak izan, ez dute ahalmen haundirik sarearen barneko elementuetan influentzia izateko. Beste sailkapen posible bat, metodoak ematen duen irteeraren arabera sailkatu daiteke. Metodo batzuk, irteera estatistikoak eskaintzen dituzte, grafiketan ikusarazi daitezkeenak; beste batzuk sarearen barneko parametroak eskaintzen dituzte[284]; badaude ezaguna den modelo baten aurka sarea konparatzen dituzten metodoak eta azkenik irteeretatik eta sarreretatik datuak ateratzen dituztenak. Azkeneko hauek oso erabilgarriak izan daitezke irudiekin lan egiten duten sareetan, sarreraren ezaugarriren bat aldaturik, ea sarea irteera egokia eskaintzen jarraitzen duen aztertu daiteke, irteeraren aldaketa noiz gerta egin den ikerturik[55].

Parametro gutxi batzuk baino gehiago dauzkan edozein modeloren ulergarritasuna oso konplexua da eta ia ezinezkoa zuzenean egiten saiatzen bada[55], [263]–[265], [282], [285]. Modelo guztiak ezin dira era parametrikoko baten ulertu, kasu gehienetan hain lotuta daude parametro guztiak beraien artean, ia ezinezkoa dela gizaki adimenerako konplexutasun guzti hori menderatzea. Benetan posible dena, era mugatu baten modeloen zati bananduak ulertzea edo era lokal batean lan egitea. Modelo konplexu baten iragarpena sarearen parametro batzuen menpe egotea era lineal edo monotono baten. Kasu honetan, sarreran aldaketa txikiak eginik, irteeran sortzen diren iragarpenen ulertze prozesu bat egin daiteke[55]. Posible da, ere, ulertze prozesu bat egitea era lokal baten datu multzo bat harturik eta globala balitz moduan tratatzea, ulertze teknika modularrak erabiliz.

Modelo bat era ulergarri baten aztertzeke hiru era nagusi daude [55], [262]:

- Aplikazioko ikuspuntutik: Modeloa bukaerako erabiltzaileengatik frogatuta izangoa. Erabiltzaile hauek frogatu nahi den gaian jakituria sakona eduki beharko dute.**
- Gizaki ikuspuntutik: Modeloa gaian adituak ez diren pertsonengatik frogatu egiten da.**
- Funtzionaltasun ikuspuntutik: Proxy automatiko baten bitartez modeloa frogatu egiten da. Ikuspuntu honen emaitzak, orokorrean, hobetoak izaten dira lehenago gizaki ikuspuntutik frogatu bada modeloa.**

3. SGD Algoritmoaren hobekuntzak

Gradiente estokastiko beherakorrak, SGD ingelesez, ahultasunak aurkezten dituen ikasketa algoritmo bat da, ala ere, oso zabaldua izanik. Le Roux et al.[286] eta Defazio et al.[287] algoritmoari memoria gehituz hau modifikatzea proposatzen dute, lehenengo kasuan, pasaden epokotako gradienteak gordez eta beraien mediaren bitartez pisu sinaptikoen zuzenketa egitea, eta bigarren kasuan, lehenengoaren eboluzio bat izanez, gradientearen eraldaketa inkrementala erabiliz. Bi kasuetan, errore baxuetara konbergentzia azkarrago bat lortzen dutela diote. Ildo berdinetik jarraituta, Johnson eta Zang[288] bariantza prediktiboaren erredukzioa erabiltzen duen aldaketa bat planteatzen dute SGD algoritmoan, ere konbergentzia azkarra lortuz Loss minimoetara. Zhuang Yang[289] gradiente konjugatuen algoritmoa eta gradiente estokastiko beherakor errekurtsiboa, lehengoaren konbergentzia abiaturaz eta bigarrenaren ikasketa jausi zuzentasunaz baliatuz, konbergentzia on bat lortzeko arrazoizko denbora kantitate baten. Antzerako era baten, Pengfei Wang eta Nenggan Zheng[290] bi metodo aurkezten dituzte emaitza minimora konbergentzia zuzen bat lortzeko gradiente estokastiko errekurtsibo asinkrono bat erabiliz. Li et al.[291] gradiente estokastiko beherakor paraleloaren aldaketa bat proposatzen dute espazio barreneko komunikazio optikoetarako. Beraien aldaketa, Hu et al.[292] zuntz optikorako proposatutako lanaren jarraipen bat da. Aldaketa bera, Loss berrelikatzea patroia bakar baten datuekin egin ordez, patroia bat baino gehiagoren datuekin elikatzen da. Le Trieu Phong eta Tran Thi Phuong[293] bete aldaketa bat proposatzen dute SGD-an, non, memoria eta konpresioa gehitu egiten zaio, sareak zehaztasun haundiagoa eta zarata txikiago jasan dezan ikasteko prozesuan. Fidagarritasunaren eta seguritatearen aldetik, Roberts eta Smith[294] erraztutako konbergentzia teoria bat aurkezten dute, bere egunen Blanchard et al.[295] proposatutakoa, non, SGD algoritmoa, Bizantzio motako erasoaren aurka indartu egiten da, banatutako ikasketa ereduetan.

Badaude beste SGD algoritmoan berean aldaketa nagusia planteatzen dituzte, Druchi et al.[184] AdaGrad algoritmoan oinarritutakoa bezala. Kasu honetan, ikasketa prozesuan ikusitako datuen geometria ikasketa prozesuan bertan efektu izatea eragiten. Honek, datuen ezaugarri komunenei ikasketa ratio txikiak izendatzen dizkie eta altuagoak arraro diren ezaugarriei. Ikerketa hau 2011-n kaleratu zenez, ideia honen inguruan beste autore batzuk ere lan egin dute[61],[296]-[300]. AdaGrad-en aldaketarik inportanteena, lehen esan den moduan, RMSProp algoritmoa da, bere aitzindarian aurkitutako ahultasunak zuzenduz[65],[81],[290],[301]-[303].

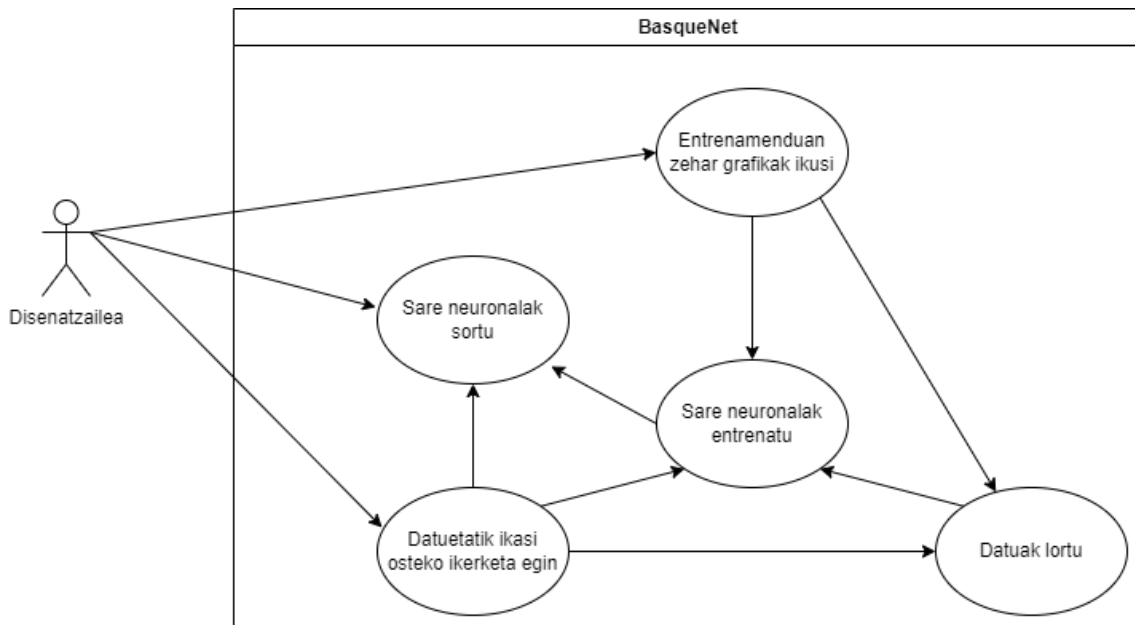
Beste autore batzuk, SGD algoritmoa beste eraldaketa eta esperimenterako erabiltzen dute. Chen et al.[70] SGD Loss paisaiaren zehar duen portaeraren inguruan ikertze sakon bat egin dute. Beraien lan osteko ondorioetan diote SGD algoritmoa jokaera super-difusoa daukala entrenamenduko lehenengo epokatan. Honek ahalbidetzen dio oso azkar igarotzea fraktal formako Loss ingurumenetik ia lauak diren minimo inguruko zonaldeetara. Gero entrenamendua igaro ahala, bere jokaera sub-difusoa izatera pasa egiten da, Loss minimora pixkanaka-pixkanaka urreratzea lau xamar den zonalde honetan. Hau egiteko SGD algoritmoaren hiper-parametro ezberdinak eta beraien influentzia sarean ikertu dituzte. Antzerako era baten, Carmina Fjellström eta Kaj Nyström[304] difusio mapa bat eraiki dute SGD-a eratzten duten elementu ezberdinekin eta Loss-aren gainazalarekin. Sun et al.[305] konputagailuko zenbaki higikorak erabiltzea, zenbaki kontinuen partez, ze efektu daukan SGD algoritmoan, gerta daitezkeen gainezkatzeekin. Zuhaitz neuronalen munduan, Varun Ojha eta Giuseppe Nicosia[306] zuhaitz neuronal bat birmoldatzen dute, neurona bakarreko zuhaitz batera eta hau SGD bitartez entrenatzen dute. Senthil et al.[307] Aladelta SGD deritzoten SGD-motako algoritmo bat eraiki dute ospitale bateko gaixoen datuak espazio eta denbora eraginkorrean gordetzeko. Era honetan, gaixotasun baten aurreikuspena erraztu egiten da.

Drop-out teknikaren aldetik, drop-out teknika aldakorrak oso erabiliak dira. Lei Jimmy Ba eta Brendan Frey[67] aitzindariak izan ziren drop-out aldakor teknikaren munduan, momentu horretan drop-out estandarrarekin lortzen zen Loss-a asko murrizuz. Li et al.[308] beraien partez, aldakortasun hau genen teorian eta gizakien burmuinean oinarritu egin dute. Beraien hitzetan, teknika honen bitartez, sareak lortu dezakeen datuen ezaugarrien erauzketa eta sailkapena hobetzen da oso. Mirzadeh et al.[309] sare baten etengabeko entrenamenduaren eta drop-outaren arteko erlazioa landu dute. Beraien lanean, froga dezakete, sare neuronal bat drop-outa erabiltzen duen heinean, erabiliko ez balu baino egonkorragoa dela, aurretik ikasitako lanak mantentzeko gai izanda, lan berriak ikasten dituen bitartean. Yuanyuan Chen eta Zhang Yi[68] drop-out modagarri urri bat proposatzen dute, oso neurona gutxi entrenatuz pauso bakoitzean. Neuronak hautatzeko algoritmoa Sigmoidea funtzioan oinarriturik dago. Azkenik, Danielle Jeune et al.[310] beraien drop-out moldakorra proposatzen dute, -zigor efektiboa, emaitza onak lortzen dituen urritasun eremuetan.

III. BasqueNet liburutegiaren garapena

1. Lortu nahi ziren objektiboak

BasqueNet planteatzen hasi zenean, Irudia 23-n ikusi daitekeen erabilera kasuetan ikusi daitekeen ideiak jarri ziren mahai gainean. Liburutegiak, beste sare neuronal sortzeko erremintek ematen ez dituzten datuak eskaini behar zituen, bai ikasketa burutzen ari zen bitartean zer nolako portaera duen sistemak ikusteko, bai gerora lortu diren datuetatik ondorioak ateratzeko. Datu hauekin, sareak izan duen portaera ikertzea lortu nahi da, berarengan eragina daukaten faktore guztiekin, edo behintzat teknikoki lortu ahal direnekin. Era honetan, problema bat ekiditeko, behar den sare neuronalik txikiena lortu nahi da, bai optimizazioaren aldetik, bai ulergarritasunaren aldetik.



Irudia 23: Erabilera kasuak.

Aurreko guztia kontutan harturik, hurrengoko baldintzak definitu dira BasqueNet-erako.

1.1. Baldintza funtzionalak

Hurrengoko lerroetan BasqueNet liburutegiarekin lortu nahi diren baldintza funtzionalen zerrenda aurkitu daiteke.

- BF1: Liburutegia erabilia edozein sakonerako geruza ugariko edozein sare sortu daiteke.**
- BF2: Liburutegia ikasketa parametroak ezartzeko ahalmena izan behar du.**
- BF3: Liburutegia geruzak definitzeko aukera eman behar du.**
 - Geruzek hasiera funtzio bat izan behar dute.**
 - Geruzek prozesu funtzio bat izan behar dute.**
 - Geruzek Atzera norantzako funtzio bat izan behar dute, gradientearen atzerantzako hedapena emateko.**
 - Geruzek beraien Jakobinoak lortzeko aukera eman behar dute.**
- BF4: Liburutegia guztiz konektaturiko geruzak definitzeko ahalmena izan behar du.**
 - Geruza hauetan sarrera eta irteera kopurua definitzeko ahalmena izan behar dute.**
 - Sarrera kopurua ezin da irteera kopuruaren menpe egon eta alderantziz.**
 - Geruza hauetan ikasketa alpha, betta eta Dropout-a ezartzeko aukera eduki behar dute.**
 - Geruza hauek sarreren aurrerantz elikadura edukitzeko aukera izan behar dute.**
 - Geruza hauek, azkeneko aldaketa deuseztatzeko aukera eman behar dute.**
 - Geruza hauek, sarrera kantitatea aldatzeko funtzio bat izan behar dute.**
 - Geruza hauek irteera kantitatea aldatzeko funtzio bat izan behar dute.**

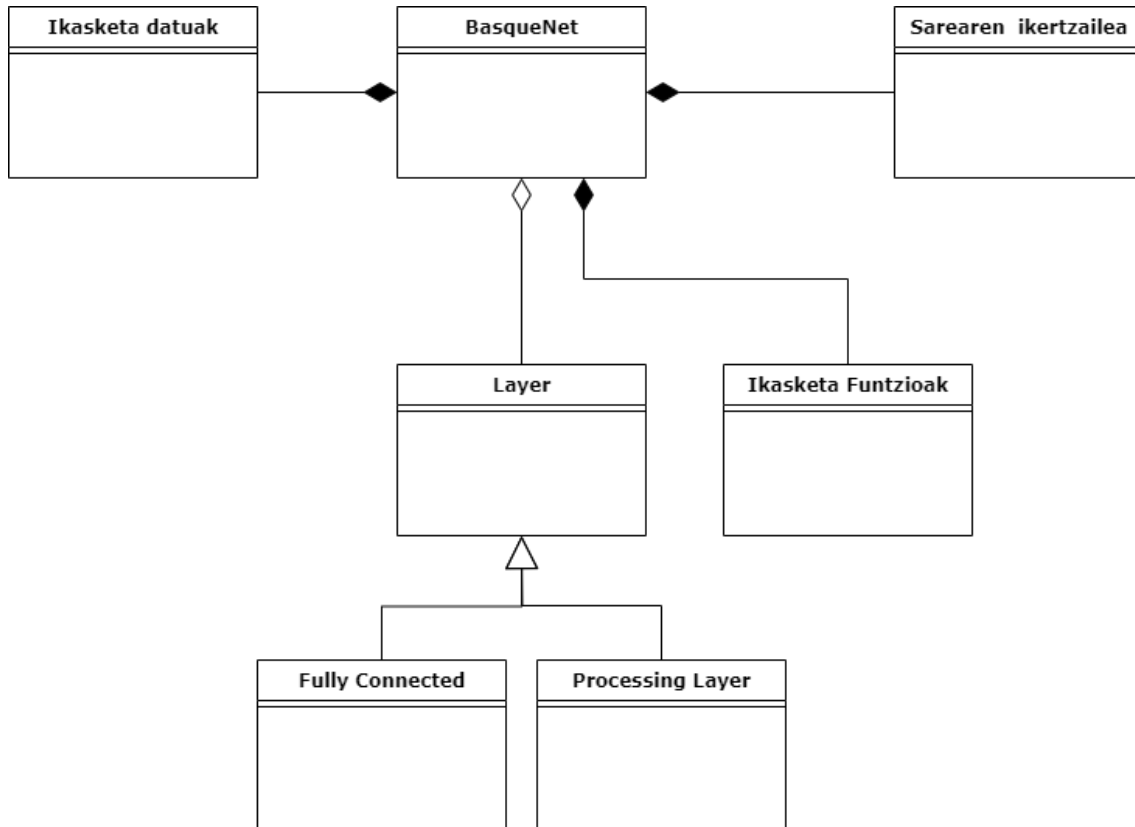
- **BF5: Liburutegia Prozedurako geruzak sortzeko aukera eman behar du.**
 - Geruza hauen definizioa era generiko baten eman behar da.
 - Geruza hauek elementu kopurua aldatzeko funtzio bat izan behar dute.
- **BF6: Prozedura geruzen artean ReLu mota egon behar da.**
- **BF7: Prozedura geruzen artean Sigmoidea mota egon behar da.**
- **BF8: Prozedura geruzen artean SoftMax mota egon behar da.**
- **BF9: Liburutegia Loss funtzio ezberdinak ezartzeko aukera eman behar du.**
 - Funtzio bakoitzaren implementazioa irteera bat emanda, bere errorea kalkulatzeko gai izan behar da.
 - Funtzio bakoitzean honen deribatua ateratzeko aukera egon behar du.
- **BF10: Loss funtzioen artean MSE hautatzeko aukera egon behar du.**
- **BF11: Los funtzioen artean Entropia Gurutzatutako algoritmoa egon behar du.**
- **BF12: Liburutegia SGD algoritmoa, ikasketa algoritmo bezala hautatzeko aukera eman behar du.**
- **BF13: Liburutegia Levenberg-Marquard algoritmoa, ikasketa algoritmo bezala hautatzeko aukera eman behar du.**
- **BF14: Liburutegia ikasketaren prozesuaren zehar ikasketa beraren datuak gorde behar ditu.**

1.2. Baldintza ez-funtzionalak

- **BEF1: Ikasketaren zehar lortzen ari diren datuak grafikoetan islatuko dira.**
- **BEF2: Ikasketaren zehar ikasitako epoka kopurua, azkeneko epoka ikasteko eman duen denbora eta azkeneko batezbesteko errorea entrenamendu patroiekin eta balidaziokoekin.**

2. Liburutegiaren arkitektura

Baldintzak definiturik hasierako arkitektura bat planteatu da liburutegiaren estrukturarako. Irudia 24-n ikusi daiteke nola banatuta dagoen. BasqueNet inplementazio bat sortzeko derrigorrezkoa da ikasketa datuak definitzea(alpha, beta, Dropout,...), erabiliko den ikasketa funtzioa(MSE, Entropia Gurutzatua,...) eta sare ikertzaile bat. Puntu horretatik aurrera, BasqueNeten geruzak sartu ahal dira, nahi beste, eta nahi den ordenan.



Irudia 24: Liburutegiaren arkitektura.

Arkitektura honetaz, nahi den malgutasuna dago, sareak, nahi den estrukturarekin ezartzeko. Geruzek, interfaze berdina aurkeztuz BasqueNeteri, hauen tratamendua goiko geruzan erraztu eta araupetu egiten du.

3. Detaileko diseinua

Detaileko diseinuaren partetik, liburutegia parte ezberdinetan bananduko da. Lehen esan den moduan, kodigo guztia Matlab Script motako objektuetara zuzendutako programaketan sortu egin da. Erabilitako Matlab erreminta, urtez urte azkeneko bertsioa lortuta egin da, 2020b bertsiotik hasita. Software bertsioa mantendu ahal izateko, GitHub-en biltegi baten bidez egin da. Era honetan, proiektua partekatzea erraztu egiten da eta ez da loturik egon ordenagailu bakar batera. Kodearen banaketan, elarrekin erlazioa duten klaseei familia bezala deritzate eta elkarrekin azalduko dira tokatzen zaien puntuan.

3.1. Layer familia

Klase familia hau sare neuronalaren muina da. Irudia 25-n ikusi daitekeenez, guztiz konektatutako geruzak du pisurik haundiena, bai parametroetan bai metodoetan. Klasez, klase azaldurik honako hau aurkitu dezakegu:

Layer klasea: Familia honetan oinarriko klasea, nondik beste klase guztiek oinordetuko duten. Ez du parametrarik aurkezten, oinorde bakoitzak bereak definituko baititu. Metodo publikoen artean honako hauek ditu:

- **Layer(name):** Metodo eraikitzailea, sortuko den kapa berriaren izena pasa behar zaio argumentu moduan.
- **process(input):** input sarrera bat emanda, geruzak sortzen duen irteera kalkulatzeko duen irteera ematen du. Bai ikasteko orduan, bai funtzionamenduaren orduan, geruzaren portaera ematen duen funtzio da.
- **backward(previousLayerBackward):** Irteeratik sarrerara errorea hedatzerakoan, erabiltzen den metodoa. Sarrera moduan, irteeratik gertuago dagoen hurrengo geruzaren backward funtzioaren irteera da. Geruza hau irteerakoa bada, ikasketa funtzioaren emaitza da.
- **jakobianosLortu():** Geruzak momentu horretan dauzkan jakobinoen balorea bueltatzen du.

FullyConnected klasea: Layer klasearen oinordea. Parametro ugari ditu, garrantzitsuenak `WeightsMatrix` eta `BiasMatrix` izanda. Klase hau, sarearen oinarria diren objektuak definitzen ditu. Hemen bere metodo nagusiak:

- **`FullyConnected(name, inputSize, outputSize, seed, dropout, residual)`:** Metodo eraikitzailea. Sortuko den objektu berriaren izena, sarrera, irteerak, Dropout probabilitatea eta irteera hondarrak sortuko duen.
- **`eraseLastModification()`:** Azkeneko epokan gordetako datuak borratzen ditu. Bere erabilera, datu gehiegi gordetzeagatik ordenagailuaren memoria ez saturatzeko erabiltzen da.
- **`resizeInputs(newSize)`:** Sarrera kopurua aldatzeko funtzioa.
- **`resizeOutputs(newSize)`:** Irteera kopurua aldatzeko funtzioa.

Funtzio hauetaz aparte, Layer klasetik oinordetu dituen `process()`, `backward()` eta `jaboianosLortu()` ere ditu.

ProcessingLayer klasea: Layer klasearen oinordea. Prozedura klase guztien oinarriko klasea. Beste prozedura klaseek hau oinordetu egiten dute. Hurrengoko hauek, klasearen metodoak dira:

- **`ProcessingLayer(name, size)`:** Metodo eraikitzailea. Sortuko den objektuaren izena eta sarrera/irteera tamaina ezartzen ditu.
- **`resize(newSize)`:** Sortutako geruzaren tamaina aldatzen duen metodoa.

FullyConnected klasearen bezala ProcessingLayer-ak Layer klasearen metodo guztiak ere dauzka.

ReluLayer klasea: ReLU prozedura funtzioa, ProcessingLayer-en oinordea. Hurrengoko metodoa inplementatzen du:

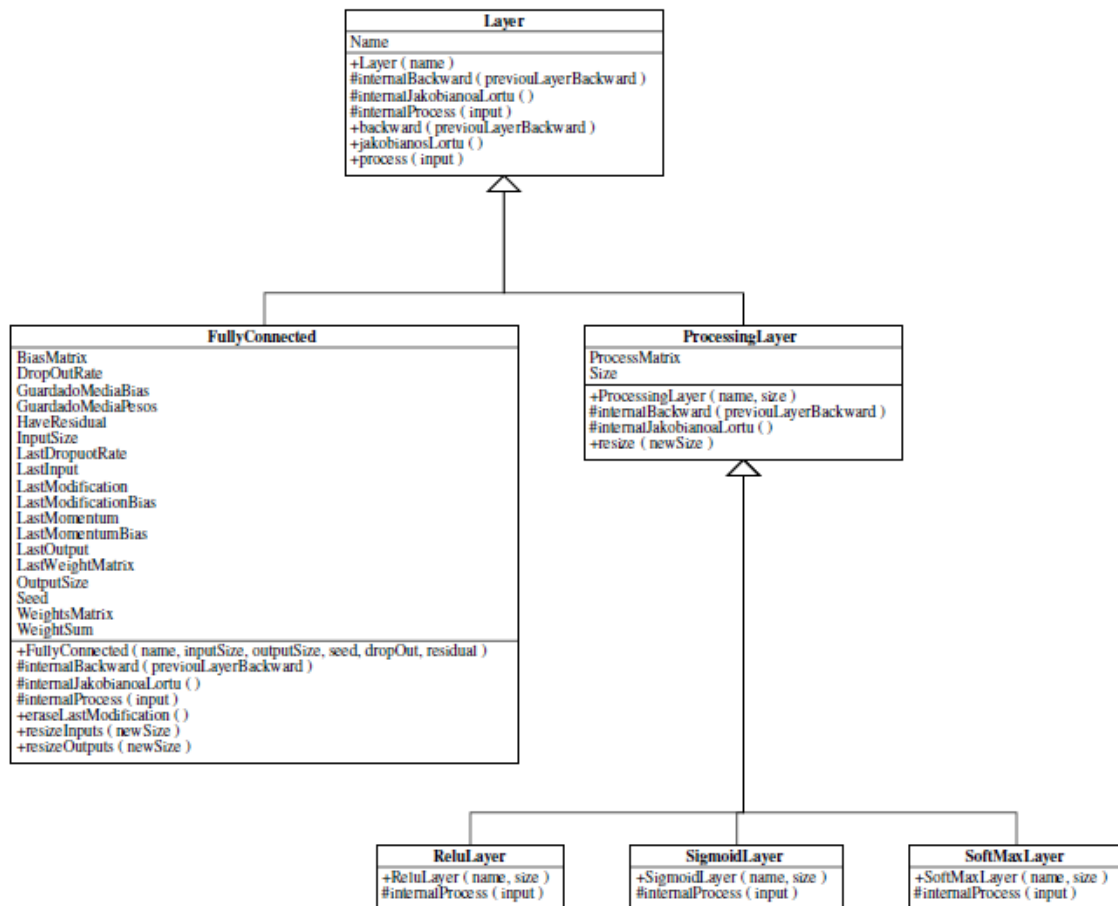
- **`ReluLayer(name, size)`:** Metodo eraikitzailea. Sortuko den ReLU geruzaren izena eta tamaina ezartzen du.

SigmoidLayer klasea: Sigmoidea prozedura funtzioa, ProcessingLayer-en oinordea. Hurrengoko metodoa inplementatzen du:

- **`SigmoidLayer(name, size)`:** Metodo eraikitzailea. Sortuko den Sigmoidea geruzaren izena eta tamaina ezartzen du.

SoftMaxLayer klasea: SoftMax prozedura funtzioa. ProcessingLayer-en oinordea. Hurrengoko metodoa implementatzen du:

→ **SoftMaxLayer(name, size):** Metodo eraikitzailea. Sortuko den SoftMax geruzaren izena eta tamaina ezartzen du.



Irudia 25: Layer klase familia.

3.2. Ikasketa funtzioen familia

Ikasketa funtzioen familia, oinarrizkoa den beste klase multzo bat da, non sarearen portaera islatu egiten da, errorea kalkulatzeko baitute eta geruzen zuzenketaren lehenengo pausua dira. Irudia 26-n familiaren estruktura ikusi daiteke:

IkasketaFuntzioa klasea: Familia honen oinarrizko klasea, beste guztiek klase onetik oinordetu egiten dute. Hauek dira bere metodo publikoak:

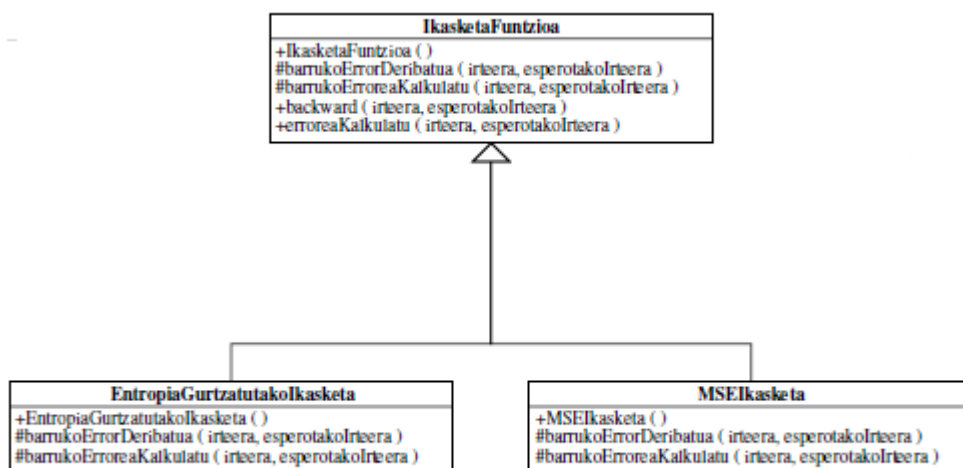
- ***IkasketaFuntzioa()***: Metodo eraikitzailea. Ez du ezer garrantzitsurik egiten objektua sortzerako orduan.
- ***backward(irteera, esperotakoIrteera)***: Geruzen zuzenketa hasteko egiten duen metodoa, sarearen azkeneko irteera, patrioiak nahi zuen irteerarekin alderatu egiten du emaitza emateko. Alderakuntza hau, bere oinorde diren klaseek inplementatzen dute.
- ***erroreaKalkulatu(irteera, esperotakoIrteera)***: Sarearen errorea kalkulatzeko duen metodoa. Aurreko kasuan bezala, oinordekoek inplementatu beharko dute metodoaren portaera.

EntropiaGurutzatutakoIkasketa klasea: IkasketaFuntzioa-ren oinorde, izen bereko ikasketa funtzioa inplementatzen duen klasea da. Berea den metodo bakarra, besteak IkasketaFuntzioa klasekoak dira, hurrengoko hau da:

- ***EntropiaGurutzatutakoIkasketa()***: Metodo eraikitzailea. Sortuko den objektuaren oinarriak ezartzen ditu.

MSEIkasketa klasea: IkasketaFuntzioa-ren oinorde, batezbesteko errore karratua inplementatzen duen klasea da. Aurreko kasuan bezala metodo publiko bakarra inplementatzen du, besteak guraso den klasetik hartzen baititu:

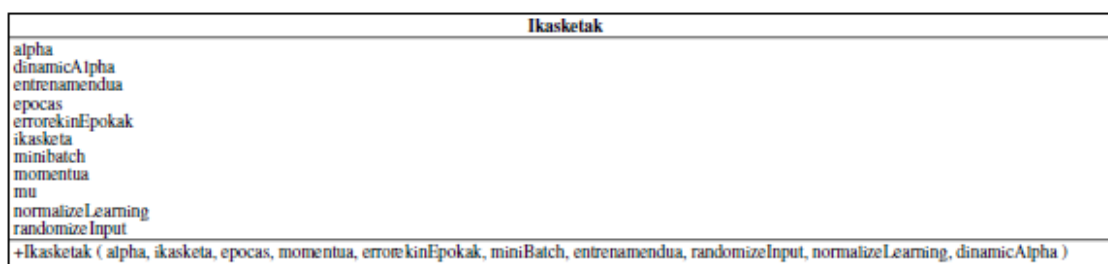
- ***MSEIkasketa()***: Metodo eraikitzailea. Sortuko den objektuaren oinarriak ezartzen ditu.



Irudia 26: Ikasketa funtzioen klaseen familia.

3.3. Ikasketa datuen klasea

Klase hau, BasqueNet erabiliko dituen entrenamenduko datuak gordetzeko erabiliko den klasea da. Klase hau metodo bakarra du, bere eraikitzailea eta parametro ugari. BasqueNet, parametro hauek erabiltzen ditu entrenamenduaren zehar. Irudia 27-n ikusi daiteke klasea.



Irudia 27: Ikasketak klasea

3.4. Sarearen ikertzailearen klasea

Entrenamenduaren zehar, sarearen portaera aztertzeko eraikitako klasea da hau. Bere metodo publikoen artean hauek daude:

- **AnalizarCapas(numLayers, basquetObj):** Metodo eraikitzailea. Geruza kopurua eta BasqueNet objektu bat behar ditu bere instantzia sortzeko.
- **analizarEvolucionRed():** Sarearen entrenamenduaren emaitzak ikertzen ditu, sarea entrenatzeari usteko aholkatuz edo sarearen estruktura aldatzeko aholkatuz.
- **incrementoEpoca():** Entrenamendu epoka berri bat hasi dela komunikatzen dion metodoa.

- **incrementoPatron():** Entrenamendua patroia berri bat sartu dela esaten zaio sare ikertzailerara.
- **registrar(LayerPostua, correccionBias, correccionPesos):** Azkeneko patroia dela eta tokatzen den geruzaren zuzenketak gordetzen ditu.
- **registrarError(error):** Azkeneko patroiaaren errorea gordetzen du.
- **registrarJacobiano(LayerPostua, Jacobiano, LastInput):** Azkeneko entrenamendu pausuko geruza baten Jakobinoa eta azkeneko sarrera gordetzen ditu.
- **registrarLoss(Loss):** Azkeneko Lossaren balioa gordetzen duen metodoa.

AnalizaCapas
CurrentEpoch
CurrentIndex
ModificadorPeso
NivelCorteError
NivelCorteLoss
regist
+AnalizaCapas (numLayers, basketObj)
+analizarEvolucionRed ()
+incrementoEpoca ()
+incrementoPatron (patron)
+registrar (LayerPostua, correccionBias, correccionPesos)
+registrarError (error)
+registrarJacobiano (LayerPostua, Jacobiano, LastInput)
+registrarLoss (Loss)

Irudia 28: AnalizaCapas klasea.

3.5. BasqueNet klasea

Azkenik, liburutegi honen muina den klasea. Irudia 29-n ikusi daiteke klasearen elementu guztiak. Hurrengoko hauek bere metodo publikoak dira:

- **BasqueNet(name):** Metodo eraikitzailea. Sarearen izena eskatzen du.
- **addLayer(newLayer):** Irteerako postuan geruza berri bat ezartzen du.
- **Ikasi(input, target, planteatu, validationIn, validationOut):** Sarearen ikasketa prozesua abiarazten duen metodoa. Sarea entrenatuko duten patroiak, sarearen ikasketa planteatu egingo den edo balidazio patroiak pasa behar zaizkio.
- **insertarPenultimo(fcl,pl):** Irteera baino postu bat atzerago bi geruza berri sartu egiten ditu, guztiz konektatutako bat eta prozedurako bat.
- **process(input):** saretik sarrera bat pasa egiten da eta honen emaitza bueltatzen du.
- **removeLastLayer():** Irteerako geruza kentzen duen metodoa.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

→ *setLearningParameters(alpha, ikasketa, epocas, momentua, errorekinEpokak, miniBatch, entrenamendua, randomizeInput, normalizeLearning, dinamicAlpha, Equilibar, DropoutAdaptativo):* BasqueNet behar dituen entrenamendu datuak ezartzen dituen metodoa.

BasqueNet
AllWeightSum Analizadur AzkenekoEpokakoModuloak AzkenekoIterazioakoModuloak columnInput DropoutAdaptativo Entrenamendua Equilibar Layers LearningParameters MediaValoresCapa Moduloak Name NumberOfLayers OraingoEpoka rowInput StopConditions
+BasqueNet (name) #azkenekoErroreazkasi (output, target, aplicarMiniBatch, elementosMiniBatch) #capasEnPolars (X, Y) #contarFullyConnected () #dropOutBN (momentoPesos, dropOutRate, epoka, LayerPostua) #epokaBatBatidatu (validationIn, validationOut) #epokaBatKasi (input, target) #equilibrar (LayerPostua, correccionPesos) #finalPlotear (out_error, target, epoca) #kalkulatutakobianoX (kapaZenbakia) #levenbergMarquard (LayerPostua, errorDeribatua) #pisuakBeriztu (errorDeribatua, partDev, LayerPostua, aplicarMiniBatch, elementosMiniBatch) #sacarModulo (partDev) #stochastic (errorDeribatua, LayerPostua) +addLayer (newLayer) +ikasi (input, target, plotear, validationIn, validationOut) +insertarPenultimo (fct, pi) +kalkulatutakobianoBektorea (kapaZenbakia) +kalkulatutakobianoBektoreaBias (kapaZenbakia) +process (input) +removeLastLayer () +setLearningParameters (alpha, ikasketa, epocas, momentua, errorekinEpokak, miniBatch, entrenamendua, randomizeInput, normalizeLearning, dinamicAlpha, Equilibar, DropoutAdaptativo)

Irudia 29: BasqueNet klasea.

4. Liburutegiarekin lan egiteko era

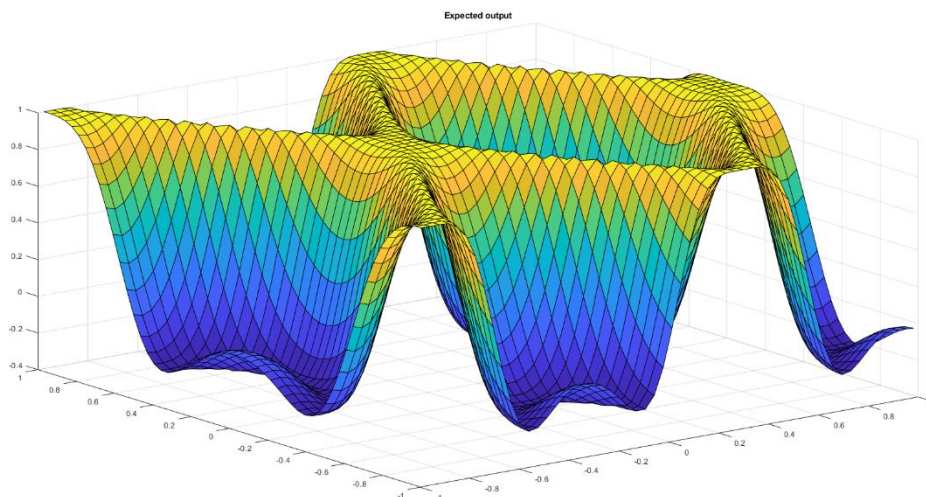
Liburutegiarekin lan egiteko beharrezkoa da Matlaben script bat sortzea ta bertatik lan egiten hastea. Lehenengo pausua, ikasketarako erabiliko diren patrioiak sortzea da. Honetarako, lan egin diren kasuetan, funtzio matematikoaren Matlabeko inplementazioa lortu eta handik patrioiak sortu. Irudia 30-n ikusi daiteke adibide bat, bertan, bai entrenamendu patrioiak, bai balidaziokoak sortzen dira. Irudia 31-n lortu nahi den formaren irudikapen bat ageri da.

```
x = (-1.9:0.05:2);
x_val = (-1.9:0.07:2);
x = x ./max(x);
x_val = x_val ./max(x_val);
y = x;
y_val = x_val;
[X,Y] = meshgrid(x);
[X_val, Y_val] = meshgrid(x_val);

F= sinc(sin(5*X) + cos(3*Y));
X_rs = reshape(X,[1,length(X(:,1))*length(X(1,:))]);
Y_rs = reshape(Y,[1,length(Y(:,1))*length(Y(1,:))]);
in=[X_rs;Y_rs];
tg = reshape(F,[1,length(F(:,1))*length(F(1,:))]);

F_val= sinc(sin(5*X_val) + cos(3*Y_val));
X_rs_val = reshape(X_val,[1,length(X_val(:,1))*length(X_val(1,:))]);
Y_rs_val = reshape(Y_val,[1,length(Y_val(:,1))*length(Y_val(1,:))]);
in_val=[X_rs_val;Y_rs_val];
tg_val = reshape(F_val,[1,length(F_val(:,1))*length(F_val(1,:))]);
```

Irudia 30: Ikasketa patroien sorkuntza.



Irudia 31: Ikasi nahi den funtzioaren forma.

Puntu horretan, beste script bat sortzea gomendatzen da, non sare neuronal bera sortzen den. Lehenengo BasqueNet objektua sortu behar da eta handik aurrera geruzak sartu egiten dira. Geruzen ordena nahi dena izan daiteke, beti kontutan izanik aurreko geruzaren irteera kopurua hurrengoaren sarreraren berdina izan behar dela, hondarrak gehituta. Irudia 32-n ikusi daiteke prozedura. Lehenik, BasqueNet sortu egiten da, bere izena emanda. Gero, ikasketaren parametroak ezartzen dira, *setLearningParameters* metodoa erabilita. Hurrengo urratsa, geruza guztiak sartzea da, bakoitzaren izena, sarrera kopurua, irteera kopurua, zenbaki aleatorioen hazia geruza horretarako, Dropout ratioa eta hondarraren erabilera.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

```
bn = BasqueNet("Adibide Sarea");  
  
bn = setLearningParameters(bn,0.01,MSEIkasketa,100, 0.01, 10, 1, "SGD",  
true,false, true, false, true);  
  
bn = addLayer(bn,FullyConnected("X5",2,100,1,0.2, false));  
bn = addLayer(bn,ReluLayer("X4", 100));  
bn = addLayer(bn,FullyConnected("X3",100,20,1,0.2, false));  
bn = addLayer(bn,ReluLayer("X2", 20));  
bn = addLayer(bn,FullyConnected("X1",20,1,1,0.2));
```

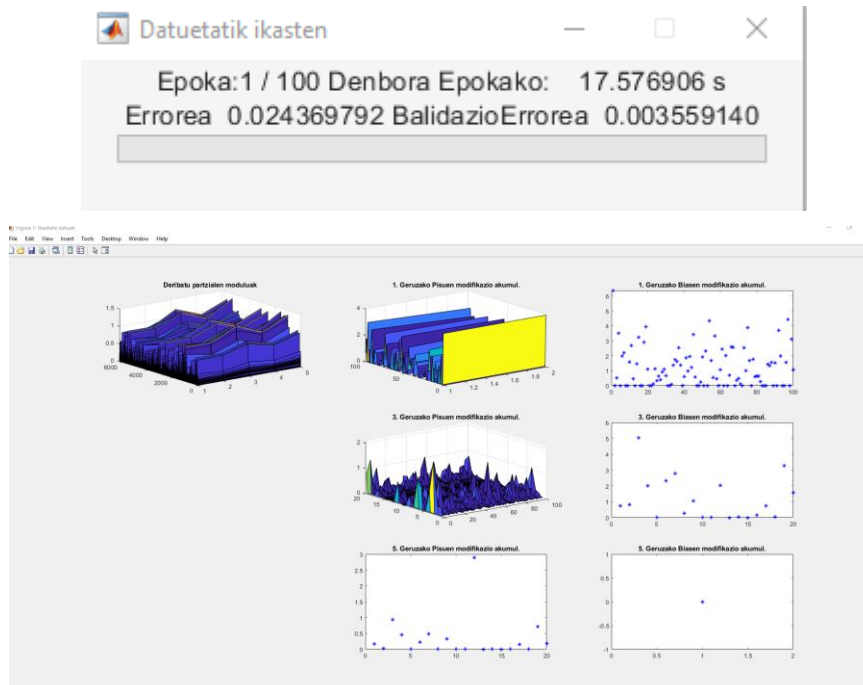
Irudia 32: Sarearen sorkuntza.

Momentu honetan, beste script bat sortzea gomendatzen da, non entrenamendua bera programatzen den. Honekin, sarea aldatu nahi bada, honek ez efekturik entrenamendu prozeduran. Irudia 33-n ikusi daiteke adibide bat.

```
close all  
[bn, error, error_val] = ikasi(bn,in,tg,false, in_val, tg_val);
```

Irudia 33: Entrenamendu scripta.

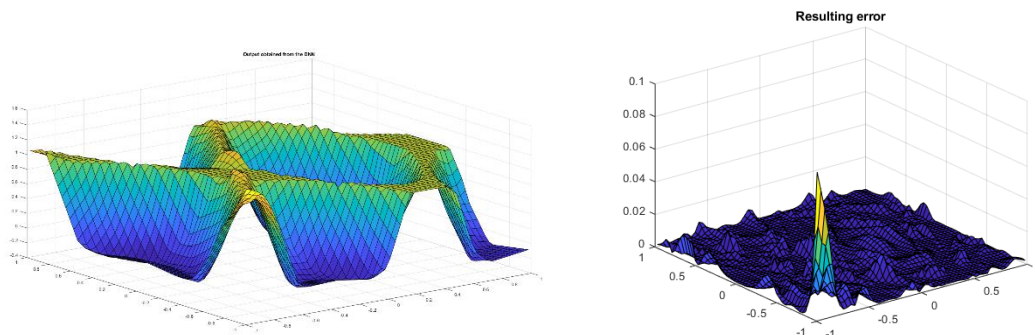
Entrenamenduan zehar, Irudia 34-n ikusten den prozesuko txantiloia eta grafikak sortzen dira, hauek, ikasketa prozesua nola dioan islatzen dute. Era berean, gain-entrenamenduagatik, ikasketa geratzen bada, azkeneko entrenatutako epokan lortutako datuak ikusi daitezke.



Irudia 34: Entrenamenduko grafikak.

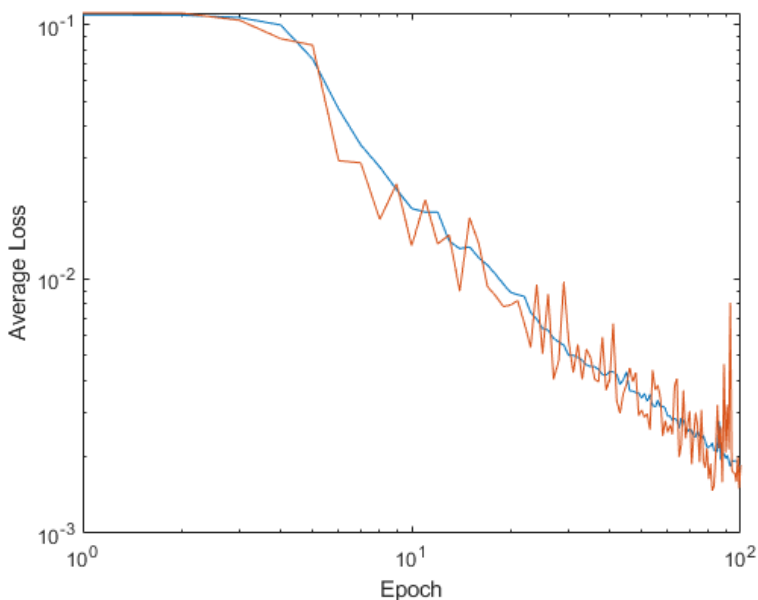
Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

Entrenamendu ostean, IV eta V ikusten diren grafikak eta datuak sortu daitezke, entrenamenduak sortu dituen erroreekin eta barnean gordetako datuekin. Kasu honetan, Irudia 35-n ikusi daitezke lortu diren forma eta errorea. Sumatzen den moduan, era orokorrean, ikasi nahi zen forma nahiko ondo barneratu du sareak, [-1, -1] puntuko zonaldean izan ezik, non errorea askotxo igo egiten da sarearen aurreikusietan.



Irudia 35: Lortutako forma eta errorea.

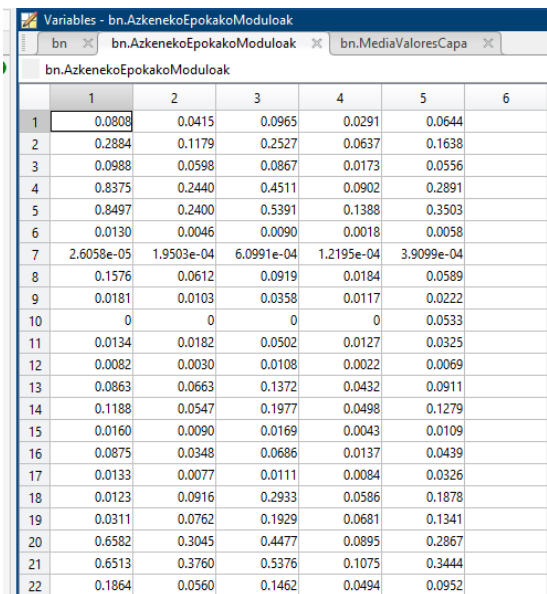
Gordetzen diren datuen artean, esanguratsu da, epokaz epoka gordetzen den sarearen Loss balioa bai ikasketa patroientzako, bai balidaziotarako. Honekin posible da, hauen bilakaera nolakoa izan den ikustea eta, gain-entrenamendua noiz hasi den ikustea. Irudia 36-n ikusi daiteke urdinez ikasketa patroien batezbesteko errorearen bilakaera eta laranjaaz balidazioena.



Irudia 36: Loss-aren bilakaera epokaz-epoka.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

Gordetzen diren beste datuen artean, azkeneko epokan, ikasitako patroio bakoitzarekiko geruza bakoitzak izandako modulua da Irudia 37-n ikusi daitekeen moduan.



	1	2	3	4	5	6
1	0.0808	0.0415	0.0965	0.0291	0.0644	
2	0.2884	0.1179	0.2527	0.0637	0.1638	
3	0.0988	0.0598	0.0867	0.0173	0.0556	
4	0.8375	0.2440	0.4511	0.0902	0.2891	
5	0.8497	0.2400	0.5391	0.1388	0.3503	
6	0.0130	0.0046	0.0090	0.0018	0.0058	
7	2.6058e-05	1.9503e-04	6.0991e-04	1.2195e-04	3.9099e-04	
8	0.1576	0.0612	0.0919	0.0184	0.0589	
9	0.0181	0.0103	0.0358	0.0117	0.0222	
10	0	0	0	0	0.0533	
11	0.0134	0.0182	0.0502	0.0127	0.0325	
12	0.0082	0.0030	0.0108	0.0022	0.0069	
13	0.0863	0.0663	0.1372	0.0432	0.0911	
14	0.1188	0.0547	0.1977	0.0498	0.1279	
15	0.0160	0.0090	0.0169	0.0043	0.0109	
16	0.0875	0.0348	0.0686	0.0137	0.0439	
17	0.0133	0.0077	0.0111	0.0084	0.0326	
18	0.0123	0.0916	0.2933	0.0586	0.1878	
19	0.0311	0.0762	0.1929	0.0681	0.1341	
20	0.6582	0.3045	0.4477	0.0895	0.2867	
21	0.6513	0.3760	0.5376	0.1075	0.3444	
22	0.1864	0.0560	0.1462	0.0494	0.0952	

Irudia 37: Patroiz-patroi geruza bakoitzaren modulua azkeneko epokan.

Geruzaz-geruza begiraturik, Irudia 38-n ikusi daiteke sarrerako geruzak gordetzen dituen datuen eredu bat. Interesgarrien diren datuen artean, geruzaren pisu sinaptikoak, "WeightMatrix", bere irteerako desbideraketekin, "BiasMatrix", daude, geruzak jaso dituen azkeneko aldaketak, "LastModification", edo azkeneko geruzaren sarrerak eta irteerak daude, "LastInput" eta "LastOutput".

Property ▲	Value
InputSize	2
OutputSize	100
WeightsMatrix	100x2 double
BiasMatrix	100x1 double
Seed	1
LastInput	[1;1]
LastOutput	100x1 double
LastMomentum	100x2 double
LastMomentumBias	100x1 double
WeightSum	[1]
DropOutRate	0.2000
LastDropuotRate	0.2000
GuardadoMediaPesos	100x2 double
GuardadoMediaBias	100x1 double
LastModification	100x2 double
LastModificationBias	100x1 double
LastWeightMatrix	100x2 double
<input checked="" type="checkbox"/> HaveResidual	0
Name	"X5"

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

Irudia 38: Sarrerako geruzan gordetzen diren datuak.

Sarearen ikertzailearen barruan sartu ezker, aurkitu daitezkeen datuen artean azkeneko epokan patroiz patroiz sortu diren datuak agertzen dira. Irudia 39-n ikusi daiteke zerrendaren hasiera, non patroien balioa dagoen, "Capa" deituriko esparruan estruktura bektore bat gorde egiten da, non geruzaz geruza patroiz horrek sortu dituen aldaketak, sortu duen Jakobinoa eta sartu dion sarrera agertu egiten den, Irudia 40-n ikusten den moduan. Datu guzti hauekin, ikasketa bukatu orduko analisi sakonak egin daitezke sareak izan dituen jokaeren inguruan. Prozedura geruzek ez dute aldaketa matrizerik, ez bait dute zuzendu daitezkeen pisu sinaptikorik lotuta.

Fields	valor	Capa	Loss
1	[-0.4000;-0.2750]	1x5 struct	-0.0203
2	[-0.4500;0.5750]	1x5 struct	-0.0516
3	[-0.2000;-0.3500]	1x5 struct	-0.0175
4	[0.7750;-0.0250]	1x5 struct	-0.0911
5	[-0.1250;0.0750]	1x5 struct	0.1104
6	[0.7250;-0.2750]	1x5 struct	0.0018
7	[0.4750;0.0500]	1x5 struct	1.2322e-04
8	[-0.5000;1.1102e-16]	1x5 struct	0.0186
9	[0.3000;0.8250]	1x5 struct	0.0070
10	[0.4500;-0.7750]	1x5 struct	0.0168
11	[-0.1250;0.8250]	1x5 struct	-0.0103
12	[1.1102e-16;0.0500]	1x5 struct	0.0022
13	[-0.9500;-0.8000]	1x5 struct	-0.0287
14	[-0.3000;0.5250]	1x5 struct	0.0403
15	[0.8500;0.4000]	1x5 struct	-0.0034

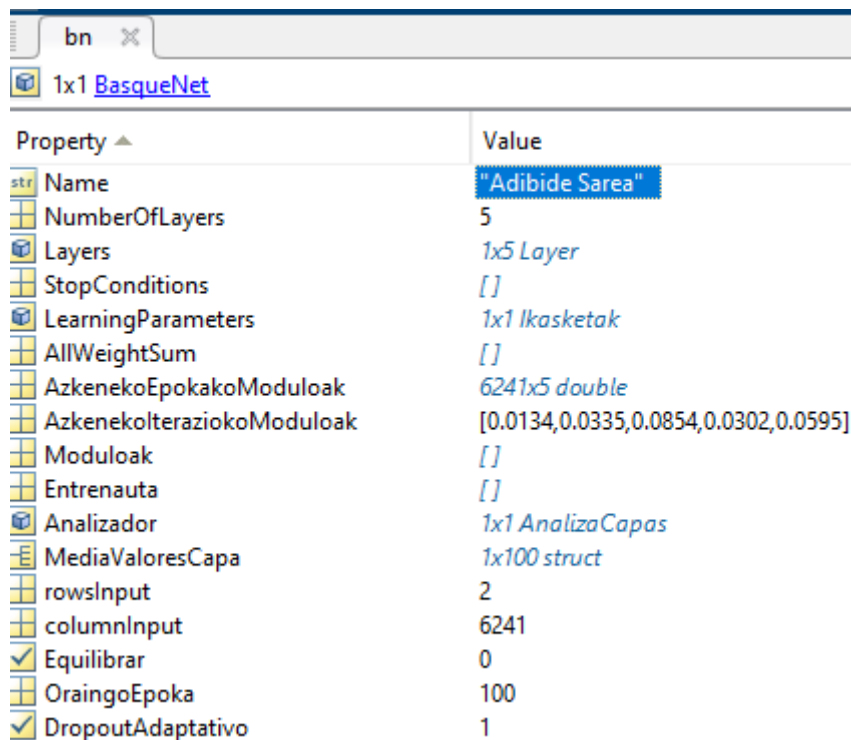
Irudia 39: Sare ikertzaile barruan gordetzen diren patroien datuak.

Fields	modificacionPesos	modificacionBias	Jacobiano	LastInput
1	100x2 double	100x1 double	100x2 double	[-0.4000;-0.2750]
2	[]	[]	100x100 double	[]
3	20x100 double	20x1 double	20x100 double	100x1 double
4	[]	[]	20x20 double	[]
5	1x20 double	-2.0299e-04	1x20 double	20x1 double
6				

Irudia 40: Azkeneko epokako lehengo patroiak sortu dituen aldaketak geruzaz-geruza eta bakoitzak jaso duen sarrera.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

Azkenik, BasqueNet estrukturan bertan gordetzen diren datuak begiraturik, Irudia 41n agertzen den moduan, honako hau ikusi daiteke. Alde batetik, izenaz aparte, sareak dituen geruza kopurua adierazten ditu eta hauek bektore batean gordetzen ditu. "LearningParameters" estruktura barruan ikasketarako erabiliko diren parametroak gordetzen dira, setLearningParameters funtzioarekin sartu diren berdinak. Sarearen ikertzailea ere ikus daiteke "Analizador" bezala agertzen delarik. Gero, gainontzeko datu guztiak, ikasketan zehar erabili diren datuak dira gerora, aurreko kasuetan bezala, sarearen portaeraren analisia egiteko balio dutelarik.



Property	Value
Name	"Adibide Sarea"
NumberOfLayers	5
Layers	1x5 Layer
StopConditions	[]
LearningParameters	1x1 Ikasketak
AllWeightSum	[]
AzkenekoEpokakoModuloak	6241x5 double
AzkenekolteraziokoModuloak	[0.0134,0.0335,0.0854,0.0302,0.0595]
Moduloak	[]
Entrenauta	[]
Analizador	1x1 AnalizaCapas
MediaValoresCapa	1x100 struct
rowsInput	2
columnInput	6241
Equilibrar	0
OraingoEpoka	100
DropoutAdaptativo	1

Irudia 41: BasqueNeteko barruko datuak.

Datu guzti hauekin, landu ezker, ikasketa prozesua nolakoa izan den ikertu daiteke. Horretarako, Matlabeko script-en bitartez, hurrengoko datuak atera daitezke adibide moduan:

- Txartoen eta hoberen ikasi diren patroien datuak ikusi daitezke eta zer portaera izan duten geruzaz-geruza, Irudia 42-n ikusi daitekeen moduan.**
- Txartoen eta hoberen ikasi diren patroien jokaera geruzaz geruza. Era honetan adibidez ikusi daitezke zer pisu sinaptiko kitzikatuta izan dira edo patroiak lerrokaturik dauden geruzaren pisuekiko. Irudia 43-n adibide bat ikusi daiteke.**

Orohar, datu hauen garrantzia eta balioa hurrengo sailetan azalduko dira. BasqueNet liburutegiak eskaintzen duen programaketa malgutasuna oso altua da eta ikusiko den moduan frogatu nahi izan den esperimendu bakoitzerako kodea alda ahal izan da era erraz baten ikuskatu nahi ziren datuak kanpora ateratzeko. Esan beharra dago, kodea bere hasierako inplementazioan epoka guztietarako gordetzen zituen, baina, honek suposatzen zuen ordenagailuko memoria kargan hain haundia zen, ikasketa blokeaturik geratzera eramaten zuela. Hori dela eta, bakarrik azkeneko epokako datuak soilik gordetzea erabaki zen.

Worst Case Pattern 1 [-0.875;-0.900]
 Partial dLoss/dX1 COS(PHI) to Total Avg.: 1.00 PHI= 0.00Degrees
 Partial dLoss/dX1 Module to Total Avg. relationship: 5232.76
 Partial dLoss/dW_X_1 COS(PHI) to Total Avg: 0.91 PHI=24.55
 Partial dLoss/dW_X_1 Module to Total Avg. relationship: 37375.09

Partial dLoss/dX2 COS(PHI) to Total Avg.: 1.00 PHI= 0.05Degrees
 Partial dLoss/dX2 Module to Total Avg. relationship: 5231.36

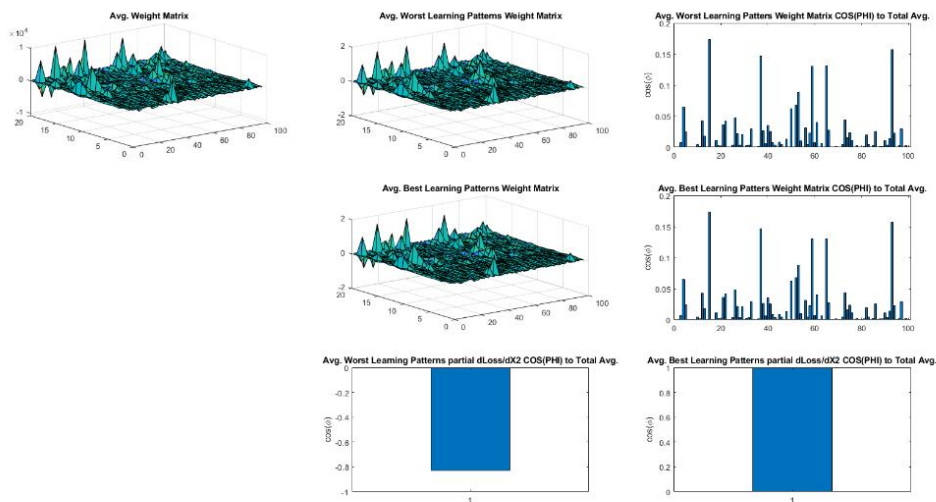
Partial dLoss/dX3 COS(PHI) to Total Avg.: -0.79 PHI= 142.33Degrees
 Partial dLoss/dX3 Module to Total Avg. relationship: 24.36
 Partial dLoss/dW_X_3 COS(PHI) to Total Avg:
 -0.45 PHI=117.01
 Partial dLoss/dW_X_3 Module to Total Avg. relationship: 63.09

Partial dLoss/dX4 COS(PHI) to Total Avg.: -0.83 PHI= 145.59Degrees
 Partial dLoss/dX4 Module to Total Avg. relationship: 0.00

Partial dLoss/dX5 COS(PHI) to Total Avg.: -0.31 PHI= 107.97Degrees
 Partial dLoss/dX5 Module to Total Avg. relationship: 28.56
 Partial dLoss/dW_X_5 COS(PHI) to Total Avg:
 0.31 PHI=72.03
 Partial dLoss/dW_X_5 Module to Total Avg. relationship: 1013.92

Partial dLoss/dIn COS(PHI) to Total Avg.: 0.85 PHI= 31.57Degrees
 Partial dLoss/dIn Module to Total Avg. relationship: 33.07

Irudia 42: Txartoen ikasi den patroiaren datuak.



Irudia 43: Txartoen ikasi diren 20 patroiaren batezbesteko jokaera.

IV. Gradiente Estokastiko Beherakorreko ikasketaren problemak

1. Elikadura zuzeneko sareak eta beraien gradiente estokastiko beherakorreko entrenamendua

I-3 puntuan esan den moduan, elikadura zuzeneko sare neuronalak, sare neuronal sakonen familia barruan sartu daitezkeen adimen artifizialeko prozedurak dira, eta beraien artean, geruza anitzeko hautemailea gehien erabiltzen dena da. Bestalde, gradiente estokastiko beherakorra, sare adimentsu hauek irakasteko metodorik zabalduenetarikoa da. Irakasketa prozesu honetan, ikasketa datu multzo bat aukeratu egingo da, datu bakoitza patroiz bezala ezagutzen da. Gradiente estokastiko beherakorra Loss funtzioa nola igarotzen den metodo bat da eta autore ezberdinez doitu egin da, kasu bakoitzeko beharretarako [62], [63], [290], [311]–[315]. Bere helburua, Loss funtzioaren minimoa aurkitzea, sarearen pisu sinaptikoak zuzenduz patroiez patroiz. Hau lortzeko, Loss funtzioaren deribatu partziala dagoen geruzarekiko atera egiten da, ta honekin pisuak zuzendu. Prozesu hau deribatu partzialen errenkada baten moduan aurkeztu daiteke, hasiera Loss funtzioaren deribatu partziala irteerarekiko jarrita eta handik aurrera, geruza bakoitzaren deribatu partziala aurreko geruzarekiko biderkatuz, dagokion geruzara iritsi arte. Hurrengoko ekuazioan honen aurkezpen formala aurkitu daiteke:

$$\begin{aligned} \frac{\partial Loss}{\partial X_n}_{1 \times N_n} &= \frac{\partial Loss}{\partial X_1}_{1 \times N_1} \cdot \left(\prod_{i=1}^{n-1} \frac{\partial X_i}{\partial X_{i+1}} \right)_{N_1 \times N_n} \\ \frac{\partial Loss}{\partial W_n}_{N_n \times N_{n+1}} &= \left(\frac{\partial Loss}{\partial X_n} \right)_{N_n \times 1}^T \left(\frac{\partial X_n}{\partial W_n} \right)_{1 \times N_{n+1}}^T \\ \frac{\partial X_n}{\partial W_{n, N_{n+1} \times 1}} &= (X_{n+1}^\mu)_{N_{n+1} \times 1} \end{aligned} \quad (3)$$

non μ saretik pasa den azkeneko patroia da, W_n geruzako pisu sinaptikoen matrizea izanik, X_n n geruza bera, lehenengoko geruza irteerakoa izanik. Matrize eta bektore bakoitzeko dimentsioa beraien aldamenean txertatu da. N_n n -garren geruzak dauzkan irteera kopurua islatzen du.

Ekuazio honetatik ateratzen den pisu sinaptikoen zuzenketa nahiko bortitza ta ezegonkorra izan daitekenez, (0, 1] bitartean dagoen ikasketa ratio bat, deritzona, aplikatzen da. Honetaz aparte, bezala ezagutzen den momentu bat aplikatzen zaio, Loss funtzioari minimo absolutura juteko joera mantenduta ikasketa prozesuan, minimo lokaletan harrapatuta gertatu partez. Momentu hau aplikatu ezean, diseinatu den sarea, eskatzen zaion funtzioa ikasteko ahulegia dela antzemango da [316]–[318]. Momentu honen balioa (0, 1] bitartean ere egongo da definiturik. Bi faktore hauek aplikaturik, pisu sinaptikoen ikasketa ekuazioa honela geratzen da:

$$\nabla W_n = -\alpha \left(\frac{\partial Loss}{\partial W_n} \right) + \beta M_n^t; M_n^{(t+1)} = M_n^t + \nabla W_n \quad (4)$$

non M_n^t momentuen matrizea den, ikasketa paso oro berriztatzen delarik. Sare baten ikasketa muturrera eramaten bada, ikasketa sasoi ugari aplikaturik, ikasketa multzotik kanpo dauden sarrerako elementuak ondo autemateko gaitasuna galtzen joango da. Ikasketa sasoiak pilatzen joan ahala sareak eskaintzen duen errorea geroz eta txikiagoa da ikasteko erabiltzen duen patroik multzuarekiko, baina, beste multzo batekiko egin ezker, iristen da momentu bat non errorea igotzen hasten dena. Hau ekiditeko, teknika bat baino gehiago aplika daiteke. Alde batetik, Batch edo Mini-Batch deritzon metodoak aplika daitezke [3], [58], [72], ikasketa patroien multzoa zatitu egiten da multzo txikiagotan, sarea Loss bitartez bakarrik zuzendurik, multzo txiki hauek pasa ostean. Honek, sarea entrenatzeko konputazio denbora jaisten du, baina sarearen konportamenduan entrenatu ostean ere eragina du, azkeneko hau txarragoa izanik. Bestalde, Dropout deitutako metodoa dago [3], [58], [72], [319], kasu honetan, Loss funtzioa saretik igartzen dabilela, pisuak zuzentzen, aldi bakoitzean, pisu portzentaje bat geruzako aldaketarik jaso gabe utzi egiten da, eta ikasketa aldi bakoitzean era aleatorio baten hautatzen dira. Era honetan, sareak era orokorrago baten ikasten du, patroik bakoitzaren detaile xumeenak jaso gabe. Azkenik, ikerkuntza kasu honetarako, patroiak sarean sartzeko ordena era aleatorio baten aldatuko da, ikasketa garaiez-garai. Honekin, sareak ezin du patroien ordena ikasi eta ez du eraginik izango sarearen konportamenduan.

2. Ikasketa algoritmoak: Aurkitutako problemak eta beraien analisia

Sare neuronalen ikasketak daukan problema nagusia patroia bat ikasteko arazoak izatea da. Hau gertatuz ezkerro, sareak ezingo ditu ondo identifikatu bere inguruan egongo diren sare neuronalaren sarrerak. Sare neuronal baten ikasketa denbora ugari kontsumitzen duen prozesu bat da eta patroia bat galtzean beste patroia multzo bat sortzea edo entrenamendu garai gehiago gehitzearen kostua ez da onargarria. Egon dira autore ezberdinak problema hau ekiditen saiatu direnak[62], [63], [192], [320].

(4)ekuazioko ∇W_n μ -garren patroia saretik pasa ostean n -garren geruzara lotuta dagoen pisu sinaptikoen gradiente matrizea da. Matrize hau normalean patroia ezberdinen datuekin aldatu egiten da eta, balore hau, ezberdina da patroia batentzako edo ere Batch barruko ∇W_n medioarekin. Guzti hau kontutan izanda, bi problema nagusi aurkitu dira ikasketa prozesuan.

Aurkitu den lehenengo problema, μ patroia baten Loss funtzioko deribatu partziala ij_n pisuarekiko, patroia guztien mediaren zeinu kontrakoa badauka; ij_n i -garren errenkadan ta j -garren zutabean dagoen W_n matrize barruko elementua da, n -garren geruzan.

$$\text{zeinua} \left(\frac{\partial \text{Loss}^\mu}{\partial \omega_{ij_n}} \right) \neq \text{zeinua} \left(\frac{\partial \text{Loss}}{\partial \omega_{ij_n}} \right) \quad (5)$$

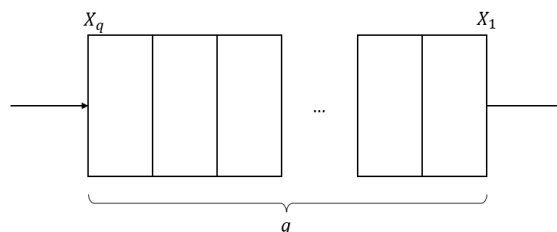
Non $\frac{\partial \text{Loss}}{\partial \omega_{ij_n}}$ Loss funtzioaren deribatu partziala ij_n -kiko patroia guztientzako media da. Kasu honetan, patroia partikular batek eskaintzen duen zuzenketa, gainontzeko patroiekin konparatuta, alderantziko zentzuan joango da, beste patroia batzuekin ikasitako elementuak ahaztera eramanez. Problema hau agertzen denean, normalean sare neuronalaren arkitekturara dago loturik, ikasketa algoritmoarekin baino. Aurkitutako bigarren problema, Loss funtzioaren deribatu partzialaren balioa 0 bilakatzea ij_n -rako.

$$\frac{\partial \text{Loss}^\mu}{\partial \omega_{ij_n}} = 0 \quad (6)$$

Kasu honetan, edo iritsi da momentu bat non Loss funtzioaren balioa txikiagia bilakatu den, edo, ReLu motako prozesamendu geruza batek 0 bilakatu du, zetorren Loss balioa negatiboa zelako. Agertzen den problema da, pisu hori ez dela zuzenduko ta berarengandik mende dauden hurrengo geruzetako pisuak ere, ez dira zuzenduko. Efektu hau gradientearen desagerketa bezala ezagutzen da. Hurrengo bi ataletan bi problema hauek sakonago aztertuko dira.

3. Gradiente desagerketaren problema

Parte honetan egingo den analisia sare neuronalaren balore Jakobiarretan oinarrituko da. Kasu generiko bat aurkezten da Irudia 44-en; non q geruzatako sare neuronal bat aurkezten da, geruza bakoitza W_n batera lotuta dagoen, n izanik geruzaren postua $[1,q]$ bitartean, X_1 irteerako geruza harturik eta X_q sarrerakoa.



Irudia 44: Sare neuronal baten irudikapen orokorra.

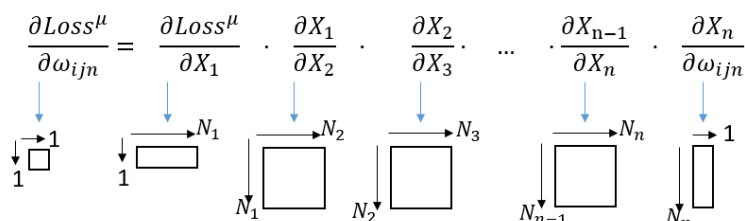
(4) ekuazioan ikusi daitezenez, W_n -ren balioa $\frac{\partial Loss^\mu}{\partial W_n}$ -en arabera kalkulatu daiteke. Pisu sinaptiko bakar batera eramanda, hurrengo ekuazioa atera daiteke, momentua aparte utzita gardentasunagatik:

$$\omega_{ijn}(t+1) = \omega_{ijn}(t) - \alpha \frac{\partial Loss^\mu}{\partial \omega_{ijn}} \quad (7)$$

non, ijn i -garren ilaran, j -garren zutabean eta n -garren geruzan dagoen pisu sinaptikoa izanda; t oraingo ikasketa prozesua izanda; ikasketa ratioa eta $\frac{\partial Loss^\mu}{\partial \omega_{ijn}}$ Loss funtzioaren deribatu partziala pisu sinaptiko horretarako μ patroiarentzat. Azkeneko elementu hau kalkulatzeko (3) ekuazio har daiteke, hurrengoko ekuazioa emanda:

$$\frac{\partial Loss^\mu}{\partial \omega_{ijn}} = \frac{\partial Loss^\mu}{\partial X_1} \cdot \frac{\partial X_1}{\partial X_2} \cdot \frac{\partial X_2}{\partial X_3} \cdot \dots \cdot \frac{\partial X_{n-1}}{\partial X_n} \cdot \frac{\partial X_n}{\partial \omega_{ijn}} \quad (8)$$

Ekuazio hau Jakobiarrek era generiko baten aztertzen ditu. Agertzen diren elementuak ikertu daitezke beraien tamainaren arabera; elementu bakoitza matrize edo bektore bat dela jakinik. Irudia 45-n ikusi daiteke irudikapen hau, N_n n -garren geruzaren irteera kopurua irudikatuz.

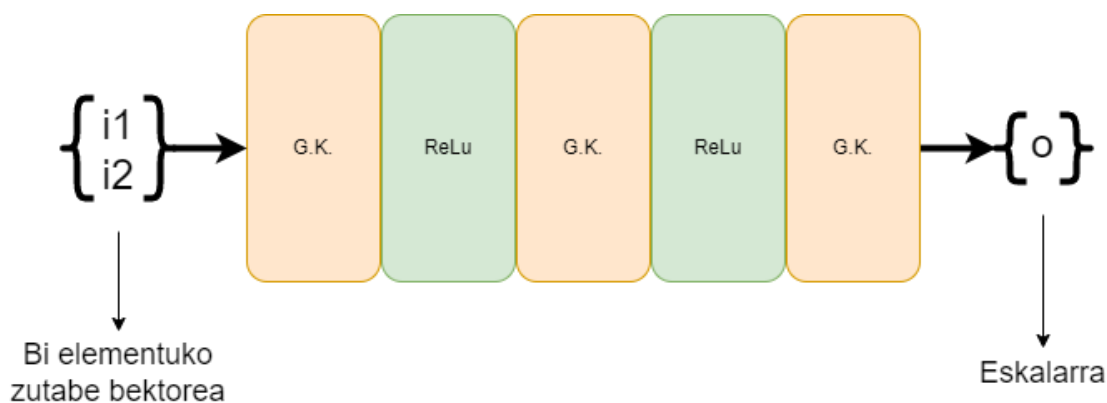


Irudia 45: (8) ekuazioaren elementuen tamainaren irudikapena.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

Hau dena kontutan izanda, moduluaren desagerpena analiza daiteke. Edozein Jakobiano matrizeren heina, (8) ekuazio barruan, zero edo oso txikia bada, informazioa galdu egingo da eta modulua pixkanaka galtzen joango da, matrize hauen erro dimentsioak oso altuak direlako. Erro dimentsio hauek oso altuak badira, matrize barruko bektore ugari balio gabeko bektoreak izango dira eta errorearen zuzenketa bertan galduko da. Hau kontutan hartuta, edozein Jakobiano matrizeren heina oso txikia suertatzen bada, errore galera bat egongo da $(n-1)$ eta n geruzen artean. Ere posible da, $\left\| \frac{\partial Loss^\mu}{\partial X_1} \right\|$ zero izatea, hau da benetan lortutako errorea 0 izatea eta ikasitako azkeneko patroia guztiz ikasi izana. Azkenik, errenkadako eskumako elementua, $\frac{\partial X_n}{\partial \omega_{ijn}}$, bektore elementu bat da. $\left\| \frac{\partial X_n}{\partial \omega_{ijn}} \right\|$ zero bada, errorearen desagerpena lego eta n -garren geruzen artean gertatu da. $\frac{\partial X_n}{\partial \omega_{ijn}}$ gehien bat zeroz osatutako bektore bat da, i -garren elementua, baina oso txikia suertatzen bada, hurrengo pausu baten elementu hau inausteko hautagai bat izango da[321]. Ala ere, hau tesi honen kanpotik geratzen den zerbait da eta hurrengo lanetarako utzi egingo da.

Ikerketa kasu honetan, Irudia 46-n agertzen den sare neuronalera erabili da. Ikusi daitekenez, sarrera bi elementutako zutabe bektore batez osatuta dago ta irteera eskalar batez. Sarea bera, 3 guztiz konektaturik dauden 3 geruzaz, GK bezala deiturik, eta 2 prozesamendu geruza zuzenketa linealeko aktibazio funtzioa erabiltzen dutenak, ReLu bezala deiturik.



Irudia 46: Ikerketarako erabilitako sarea.

(8)-en proposatzen den analisia aurrera eramateko, guztiz konektatutako geruzetan jarriko da atentzioa. Hau honela izango da, sarearen ikasteko gaitasuna bertan baitago. Bostgarren geruza, sarrerako geruza, ikertzeko, hurrengoko ekuazioa lortu egiten da:

$$\frac{\partial Loss^\mu}{\partial \omega_{ij5}} = \frac{\partial Loss^\mu}{\partial X_1} \cdot \frac{\partial X_1}{\partial X_2} \cdot \frac{\partial X_2}{\partial X_3} \cdot \frac{\partial X_3}{\partial X_4} \cdot \frac{\partial X_4}{\partial X_5} \cdot \frac{\partial X_5}{\partial \omega_{ij5}} \quad (9)$$

Lau Jakobiano matrize aztertu beharko dira, bertan informazio galera gerta bait daitekeelako. Matrizeak $\frac{\partial X_1}{\partial X_2}, \frac{\partial X_2}{\partial X_3}, \frac{\partial X_3}{\partial X_4}, \frac{\partial X_4}{\partial X_5}$ dira, ta nolako efektua daukaten $\frac{\partial X_5}{\partial \omega_{ij5}}$ gainen aztertuko da. Hirugarren geruzatik hazten bada ikerketa honako ekuazioa gertuko da:

$$\frac{\partial Loss^\mu}{\partial \omega_{ij3}} = \frac{\partial Loss^\mu}{\partial X_1} \cdot \frac{\partial X_1}{\partial X_2} \cdot \frac{\partial X_2}{\partial X_3} \cdot \frac{\partial X_3}{\partial \omega_{ij3}} \quad (10)$$

Kasu honetan bi Jakobiano matrize erabiliko dira, $\frac{\partial X_1}{\partial X_2}$ eta $\frac{\partial X_2}{\partial X_3}, \frac{\partial X_3}{\partial \omega_{ij3}}$ gainen daukaten efektua aztertuko da. Azkenik, irteerako geruza aztertzeke, bakarrik bektore elementuak edukiko dira aztertzeke $\frac{\partial Loss^\mu}{\partial X_1}$ eta $\frac{\partial X_1}{\partial \omega_{ij1}}$. Hau hurrengoko ekuazioan ikusi daiteke:

$$\frac{\partial Loss^\mu}{\partial \omega_{ij1}} = \frac{\partial Loss^\mu}{\partial X_1} \cdot \frac{\partial X_1}{\partial \omega_{ij1}} \quad (11)$$

4. Gradientearen aurkako zentzua

2-garren puntuan esan den bezala, posible da gradiente estokastiko beherakorreko ikasketa prozesua arazoak izatea, μ patroia baterako ikasketa bektorearen zentzua aurkakoa izatea gainontzeko patroienarekin alderaturik. Egoera hau gertatzen denean, patroia ikasterako orduan beste patroietan ikasitako informazioaren galera bat gertatuko da. Zentzu ezberdintasun hau neurtzeko, patroien $\frac{\partial Loss^\mu}{\partial W_n}$ matrizeak hartu dira eta bektore bilakatu dira, matrizearen errenkadak, bata bestearen atzetik jarrita, errenkada bakarreko bektore bat osatuz. Honekin, produktu eskalarra erabilia, patroia bakoitzaren angeluaren, φ -ren, kosinua patroia guztien mediarekiko kalkulatu daiteke.

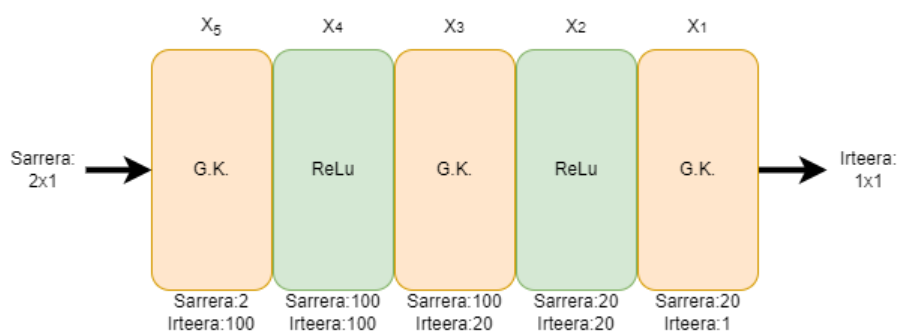
$$\cos(\varphi) = \frac{\frac{\partial Loss^\mu}{\partial W_n} \cdot \frac{\partial Loss}{\partial W_n}}{\left| \frac{\partial Loss^\mu}{\partial W_n} \right| \left| \frac{\partial Loss}{\partial W_n} \right|} \quad (12)$$

3 kasu aurkitu daitezke kosinu honen arabera. Lehenengo kasua, kosinuaren seinua positibo ta 1etik gertu badago. Kasu honetan, patroia, sarearen ikasketan efektu positiboa edukiko du, sare barruan dagoeneko informazioa gehituz ta honen errorea gutxiagotuz. Bigarren kasua, kosinua zerotik gertu egongo da. Honako honetan, sare ez gai patroia hau ondo ikasteko ta bere Loss funtzioaren igarotzea saretik ez du efektu haundirik edukiko. Kosinu honen balioa negatiboa bada, sareak informazio galera bat edukiko duela

esango du. Azkeneko kasua, kosinuaren balioa negatibo bada ta -1-etik gertu, da. Kasu honetan, informazio galera bortitza izango da, batez ere patroia $\frac{\partial Loss}{\partial W_n}$ modulua haundia bada, patroia guztien mediakoarekin alderaturik. Kasu honi patroia zapaltzaile moduan ezagutzen zaio.

5. Garatutako Matlab liburutegiaren laguntzaz egindako frogaketak

Sare neuronal xume bat eraiki da gartutako Matlabeko liburutegiarekin. Sare hau 5 geruzatan datza, 3 guztiz konektaturiko motatakoak eta bi prozedurakoak, aktibazio funtzioa ReLu izanik. GK-P-GK-P-GK, eskema jarraitzen du. Sarearen sarrera, 2 zenbaki flotatzailez ostatutako zutabe bektore bat onartzen du ta irteeran zenbaki flotatzaile eskalar bat sortzen du. Erabiltzen den Loss funtzioa, batezbesteko errore karratua da, bere erako funtzioetan komunenetakoa izanik. Irudia 47-n ikusi daiteke sare honen forma.



Irudia 47: Sortutako sare neuronalaren deskribapena.

Sarea entrenatzeko 6241 patroieko multzo bat erabiliko da, 3136 balidaziorako multzoan. Erabiliko de n entrenamendu algoritmoa SGD izango da, 100 epokatan zehar hedatuko dena. Patroien ordena epokaz epoka era aleatorio baten nahasita, sarea patroien ordena ikasteaz ekiditeko. eta 0,01 balorean jarri, Dropout-a %20-ra ezarriz. Entrenamenduko azkeneko epokan, suposatuz epoka honetan patroiek ez dutela aldaketa haundirik ekarriko sarera, bakarrik txartoen ikasi diren patroiek aldaketa haundiak egingo nahiko dituztela, patroia bakoitzaren Jakobiano guztiak gorde egingo dira. Entrenamendua bukatuz, patroiak berriz pasa egiten dira saretik, irteera hartuta ta konparatuta lortu beharreko irteerarekin sailkapen bat egin da, hoberen ikasi diren 20 patroiak ta txartoen ikasi diren 20ak harturik. Lehen esan den moduan, patroia hauen Jakobino guztiak edukirik, produktu eskalarraren bitartez hauen modulu erlazioa eta angeluaren kosinua kalkulatu egin da patroia guztien batezbestekoarekiko. Hurrengoko ataletan frogatu diren erreferentzia funtzioen esperimenduak agertzen dira. Lehenengoa, mexikar kapela

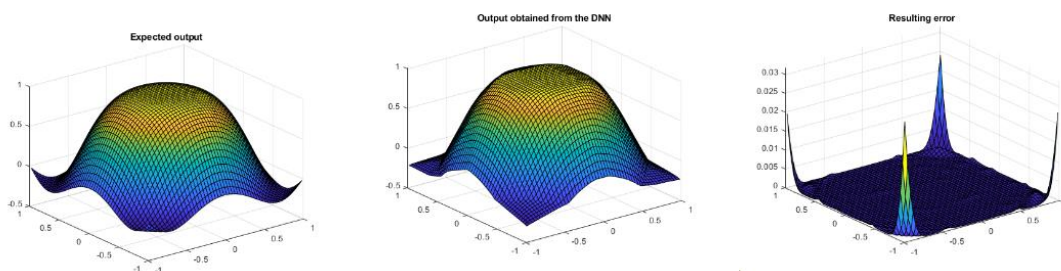
deiturikoa, ikasteko xamurra den funtzio bat da, non proba guztiak martxan jarri dira. Hurrengoko erreferentziak Surjanovic eta Bingham-en web -orritik ateratakoak dira[322].

5.1. 1. Esperimentua, Mexikar kapela

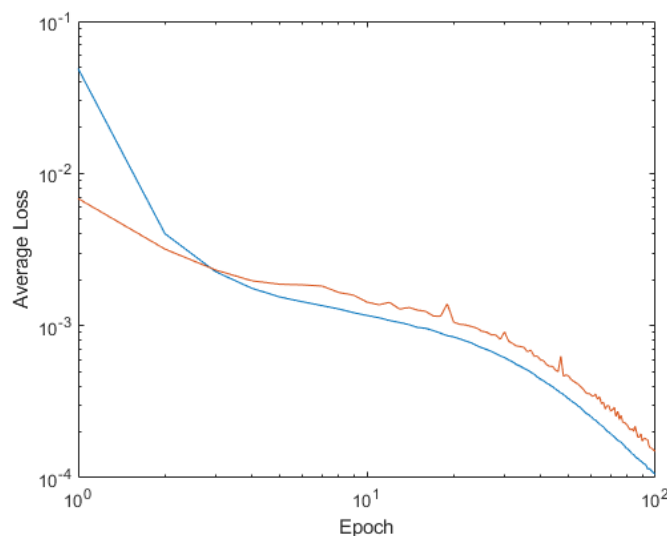
Esperimentu honetarako erabilitako helburu funtzioa, patrioak saretik pasatzean, sareak eman beharko lukeen irteeraren forma, hurrengoko hau da:

$$z = \text{sinc}(x^2 + y^2); x, y \in [-0.95, 1] \quad (13)$$

Non, x eta y sarrerako bektorearen bi elementuak dira eta z sarearen irteera. Irudia 48-n ikusi daitezke entrenamendu honen emaitzak. Irudia 49-n ikusi daiteke entrenamenduan zehar izan duen batezbesteko Lossaren balioa, urdinez ikasketarako patrioiek sortu dutena eta laranja balidazioko patrioiek lortu dutena. Garbi ikusi daiteke nola bere joera beherakorra den, maldan behera joanda, honek esan nahi du, sarea nahi den modeloa ikasten ari dela. Era berean, balidazioko patrioen Loss-a antzerako eran jaisten juteak, sarea gehiegizko entrenamendura heldu ez dela egiaztatzen du eta oraindik posible izango zela sarea entrenatzen jarraitzea.



Irudia 48: Entrenamenduaren emaitzak.



Irudia 49: Epokaz-epoka batezbesteko Loss balorea.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

Entrenamendua bukatu ostean, hobeto eta txartoen ikasitako patroien listak egin dira. Gainera, $\frac{\partial Loss}{\partial X_n}$ batezbestekoa geruzako atera egin da. Taula 1-n ikusi daiteke txartoen ikasi den patroiaaren kosinu baloreak eta moduluen erlazioak geruzaz geruza. Ikusi daitekenez, irteerako geruzetan, X_1 eta X_2 , kosinua alderantzizko norantza hartzen du eta oso balore altua moduluen erlazioan. Honek esan nahi du, informazio asko galdu behar duela sarea geruza horietan patroia hau ikasi ahal izateko.

$[-0.950; -0.950]$ Patroia	$\cos(\varphi)$	$\left \frac{\partial Loss^\mu}{\partial X_n} \right / \left \frac{\partial Loss}{\partial X_n} \right $
$\frac{\partial Loss^\mu}{\partial X_1}$	-1.00	4667.86
$\frac{\partial Loss^\mu}{\partial X_2}$	-1.00	4668.63
$\frac{\partial Loss^\mu}{\partial X_3}$	0.55	186.20
$\frac{\partial Loss^\mu}{\partial X_4}$	0.67	0.01
$\frac{\partial Loss^\mu}{\partial X_5}$	0.55	180.21
$\frac{\partial Loss^\mu}{\partial Sarrera}$	0.75	263.88

Taula 1: Errore haundiena duen patroia, geruzaz geruza batezbestekotik alderatzen duen angeluaren kosinua eta moduluen erlazioa.

Hoberen ikasi diren patroiak ostera, gehien bat batezbestekoarekin lerrokaturik dauden patroiak dira. Hala ere, Taula 2-en ikusi daitekenez, hoberen ikasitako patroiak ere kontrako norantzako geruzak aurkezten ditu, honek esan nahi du, nahiz eta patroia ondo ikasi izana, beste patroien informazioa galdu behar da hau ondo ikasi ahal izateko. Ere, aipagarria da, moduluen erlazioa hoberen ikasten diren patroietan batezbestekoarekin alderatura beti ere askoz txikiagoak dira, txartoen ikasitakoekin konparaturik.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

[0.525; -0.425] Patroia	$\cos(\varphi)$	$\left \frac{\partial Loss^\mu}{\partial X_n} \right / \left \frac{\partial Loss}{\partial X_n} \right $
$\frac{\partial Loss^\mu}{\partial X_1}$	1.00	35.73
$\frac{\partial Loss^\mu}{\partial X_2}$	1.00	35.73
$\frac{\partial Loss^\mu}{\partial X_3}$	-0.90	1.76
$\frac{\partial Loss^\mu}{\partial X_4}$	-0.99	0.00
$\frac{\partial Loss^\mu}{\partial X_5}$	-0.68	2.72
$\frac{\partial Loss^\mu}{\partial Sarrera}$	-0.99	8.70

Taula 2: Errore txikiena duen patroia, geruzaz geruza batezbestekotik alderatzen duen angeluaren kosinua eta moduluen erlazioa.

Efektua hobeto ikusteko, $\frac{\partial Loss}{\partial w}$ erabiliko da μ patroia-aren ta batezbesteko jokaeraren artean. Aldaketa honekin patroiek daukaten zuzenketa joera geruzako islatu egiten da. Taula 3 eta Taula 4-n ikusi daiteke txartoen ikasi den patroia-ren zuzenketak eta hoberen ikasi denarenak. Garbi dago, txartoen ikasi den patroiak oso gutxi dagoela lerrokaturik batezbesteko ikasketarekiko, batez ere sarrerako geruzan, X_5 , non kosinuaren balioa 0 den, influentzia gabeko patroia bat dela esanez. Hobeto ikasi dena ikusirik, ostera, irteerako geruzarekin, X_1 , ia guztiz lerrokaturik dago, naiz eta gainontzeko geruzekin kontrako norantza eduki. Erdiko geruzan, X_3 , kontrako norantza bortitza da, esanez beste patroia batzuen informazioa galdu behar duela geruza horrek patroia hau ikasi ahal izateko.

[-0.950; -0.950] Patroia	$\cos(\varphi)$	$\left \frac{\partial Loss^\mu}{\partial W_n} \right / \left \frac{\partial Loss}{\partial W_n} \right $
$\frac{\partial Loss^\mu}{\partial W_1}$	-0.17	2976.34
$\frac{\partial Loss^\mu}{\partial W_3}$	0.32	437.15
$\frac{\partial Loss^\mu}{\partial W_5}$	0.00	6847.99

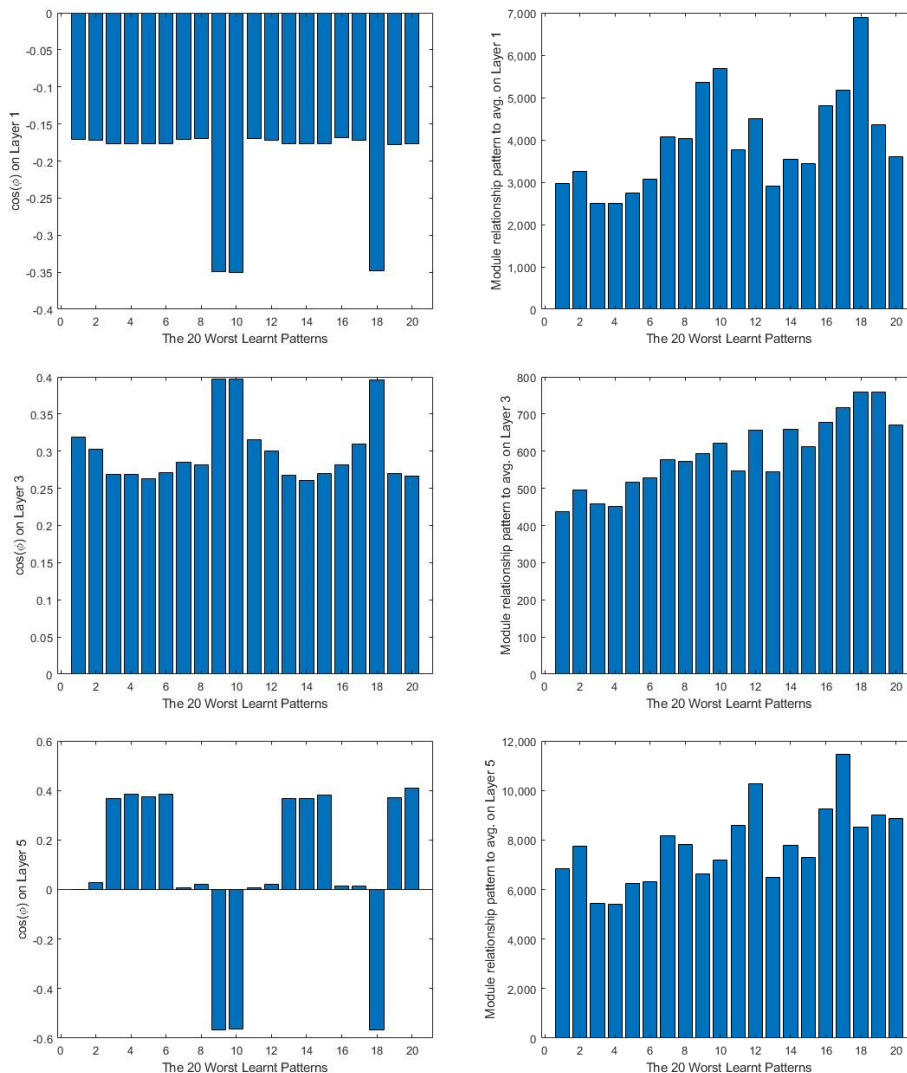
Taula 3: Errore handiena duen patroia-aren geruzen zuzenketaren kosinua ta modulu erlazioa batezbestekoarekiko.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

[0.525; -0.425] Patroia	$\cos(\varphi)$	$\left \frac{\partial Loss^\mu}{\partial W_n} \right / \left \frac{\partial Loss}{\partial W_n} \right $
$\frac{\partial Loss^\mu}{\partial W_1}$	0.98	46.34
$\frac{\partial Loss^\mu}{\partial W_3}$	-0.7	1.94
$\frac{\partial Loss^\mu}{\partial W_5}$	-0.07	51.87

Taula 4: Errore txikiena duen patroia geruzen zuzenketaren kosinua eta modulu erlazioa batezbestekoarekiko.

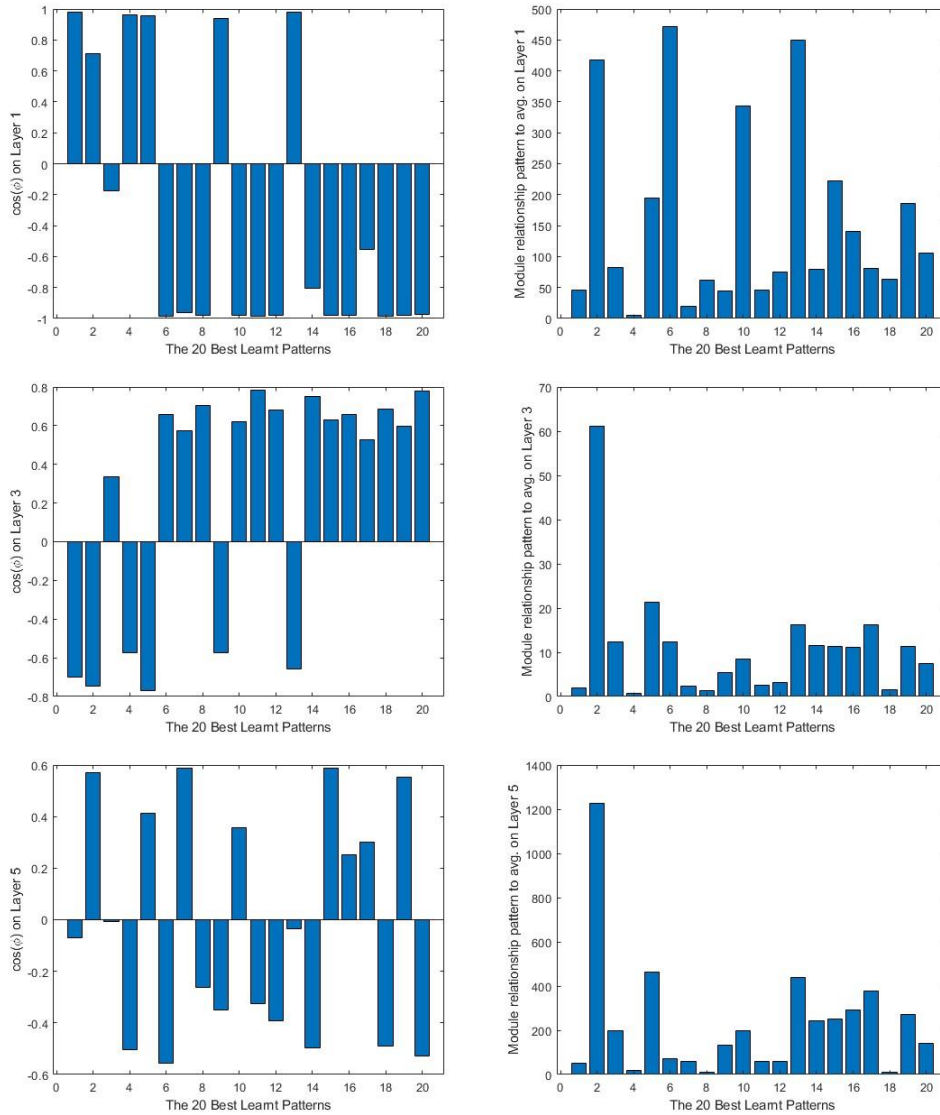
Era generikoago baten ikusteko, Irudia 50-n txartoen ikasi diren 20 patroiak daude. Ikusi daitekenez, irteerako geruzan, kosinu negatibo bat aurkezten dute ta modulu erlazio handia, honek esan nahi du patroia guzti hauek geruza honetan ikasi ahal izateko, beste patroia ugariren informazioa galdu beharko dela hori lortzeko. Sarrerako geruzan ostera, patroia hauen portaera oso aleatorioa da, kosinuak zerotik gertu mantendurik ta moduluen arteko erlazioa oso altua izanda; kasu honetan, patroiek ez dutela geruza honetan influentzia haundirik esan nahi du.



Irudia 50: Txartoen ikasi diren 20 patroien zuzenketa kosinua ezkerreko zutabeen eta $\frac{\partial \text{Loss}^H}{\partial W_n}$ -ren moduluen erlazioa batezbestekoarekiko eskuman.

Bestalde, Irudia 51-n hobeto ikasi diren 20 patroiak islatzen dira. txarto enekoekin konparaturik, kosinuaren balore absolutua askoz ere lerrokatuagoa da, bi norantzetan eta moduluen erlazioa askoz txikiagoa da. Aurreko kasuan bezala, sarrerako geruzak du portaera aleatorioena.

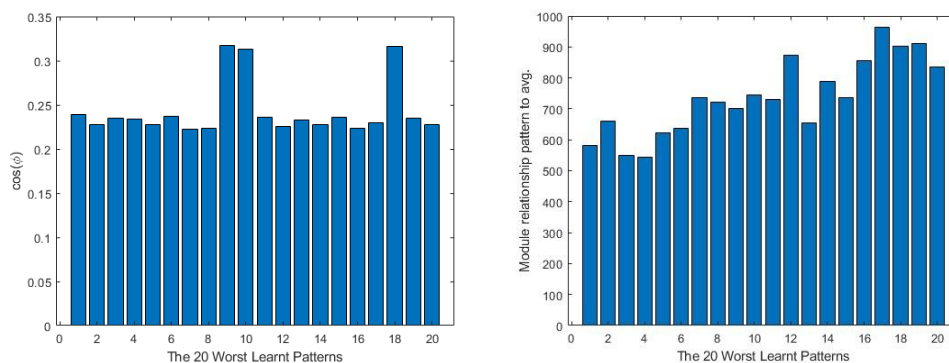
Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak



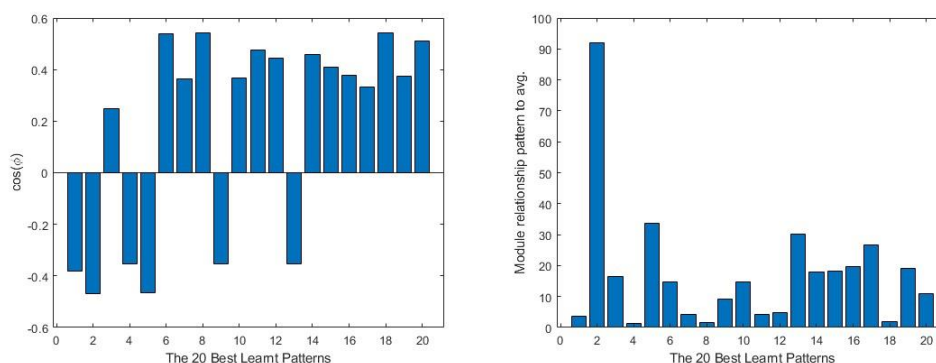
Irudia 51: Hoberen ikasi diren patroien zuzenketak sarerako, ezkerreko zutabean kosinua edukirik eta eskumakoan $\frac{\partial Loss^H}{\partial W_n}$ -ren moduluaren erlazioa batezbestekoarekiko.

Azkenik, hiru geruzen $\frac{\partial Loss}{\partial W_n}$ bat egin da, beraien lerroak bata-bestearen atzean jarrita, errenkada baka baten moduan. Hau eginda, ikusi daiteke era zehatzago baten, nola hoberen eta txartoen ikasi diren patroien artean. Irudia 52 eta Irudia 53-an ikusi daiteke ezberdintasun hau.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak



Irudia 52: Txartoen ikasi diren patroien zuzenketak sarerako, era bildu baten. Ezkerreko grafikaren kosinua egonik eta eskuman $\frac{\partial \text{Loss}^H}{\partial W_n}$ -ren modularen erlazioa batezbestekarekiko.

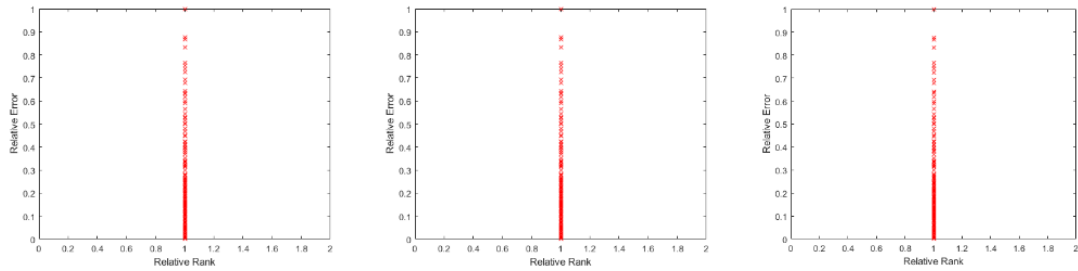


Irudia 53: Hoberen ikasi diren patroien zuzenketak sarerako era Bildu baten. Ezkerreko grafikaren kosinua egonik eta eskuman $\frac{\partial \text{Loss}^H}{\partial W_n}$ -ren modularen erlazioa batezbestekarekiko.

Gradientearen desagerketaren partetik, Jakobinoak hartu dira eta haien heina kalkulatu egin da, eduki dezaketen heina maximoagatik zatituz, hein erlatibo bat lortuz. Era berean, patroei bakoitzaren errorea, totalen kalkulatu den errore maximoarekin zatitu egin da errore erlatibo bat lortzeko. Biak erlazionaturik, hurrengoko hau lortu da:

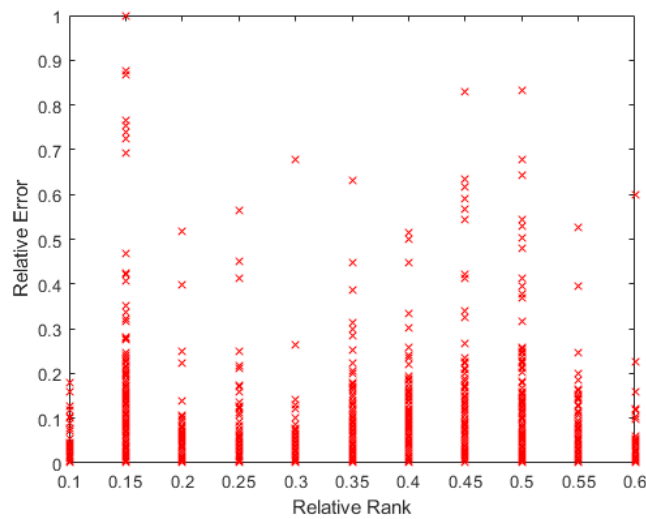
- Guztiz konektaturiko geruzetan, heina beti balore maximora mantendu egiten da patroei guztientzako. Heina hau, sarrera kopuruak eta irteera kopuruak konparatuta, bien arteko txikienaren balore berdina du. Hau Irudia 54-n ikusi daiteke.
- Prozedura geruzetan, ostera, heina ezberdinak agertzen dira patroei ezberdinetarako.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak



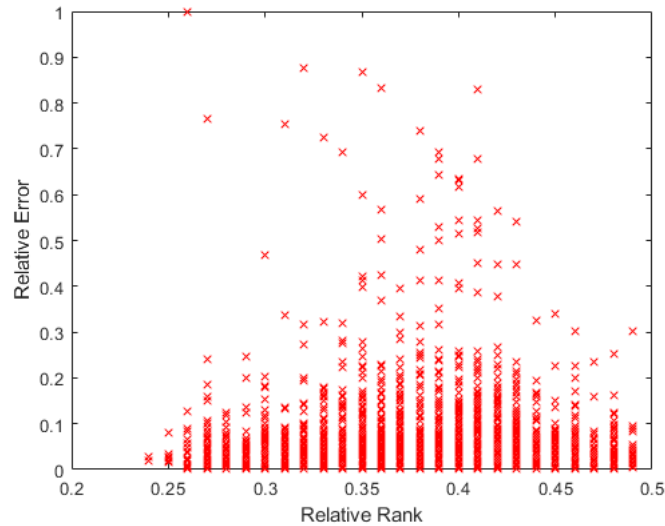
Iruia 54: Patroien banaketa guztiz konektaturik dauden geruzetan, errore erlatibo eta heina erlatiboaren arabera.

Iruia 55- ikusi daiteke nola X_2 geruzan iristen den heinarik haundiena 0.6 da, 20×20 matrize bat izanda, hori 12ko hein bat ematen du. Patroien banaketaren partetik ez dago banaketa eredu partikularrik sumatzen.



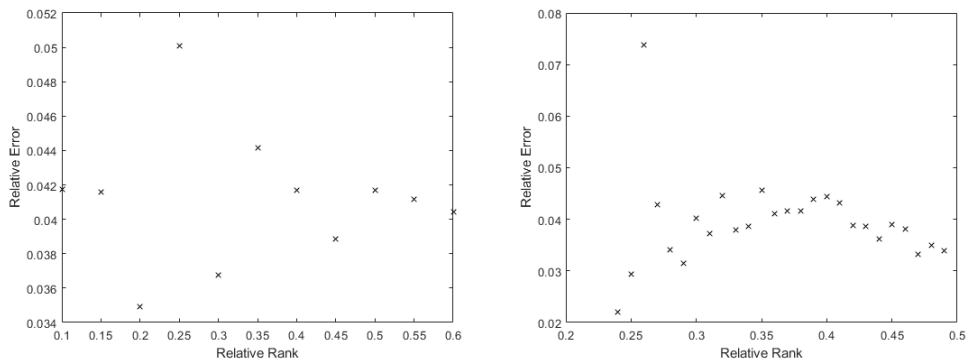
Iruia 55: Patroien banaketa, errore erlatibo eta heina erlatiboaren arabera X_2 geruzan.

Iruia 56-n, X_4 geruzarako patroien banaketa agertu egiten da. Kasu honetan, heina erlatiboa ez da 0.5 iristen, geruzaren dimentsioak harturik 50 beherako heinak aurkitu egiten dira. Aurreko kasuan bezala ez da banaketa eredu partikularrik sumatzen.



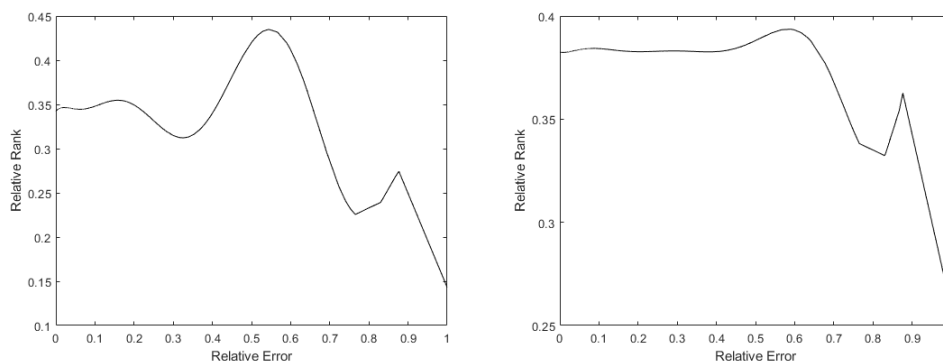
Irudia 56: Patroien banaketa, errore erlatibo eta heina erlatiboaren arabera X_4 geruzan.

Errorearen eta heinaren artean erlaziorik, ez dagoela ziurtatzeko, batezbesteko errorea atera heina bakoitzarekiko. Irudia 57-n, ikusi daiteke ez dagoela heina txikien eta errore haundien arteko loturarik, baina heina handitu ahala joera bat errorea jaisteko.



Irudia 57: Batezbesteko errorea heina erlatiboarekiko. X_2 ezkerrean eta X_4 eskuman.

Azkenik, eredia aldaturik, heina errorearen funtzio bat den frogatzeko, 9-garren mailako erregresio polinomiko bat erabili da X_2 eta X_4 geruzetan. Irudia 58-n ikusi daiteke nola bi geruzetan, heina joera beherako bat daukala errorea handitu ahala. Errore ertainetan agertzen den maximoa, batez ere X_2 -n, kontrako norantza duten patroiegatik etorri daiteke. Patroi hauek informazio asko dakarte baiian beste patroiekiko kontrako norantzan.



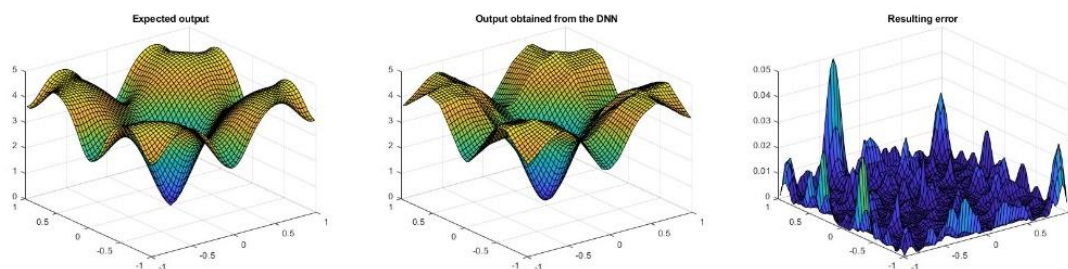
Irudia 58: Heina erlatiboaren eta errore erlatiboaren arteko erregresio polinomikoa X_2 ezkerren eta X_4 eskuman.

5.2. 2. Esperimentua, Ackley Funtzioa.

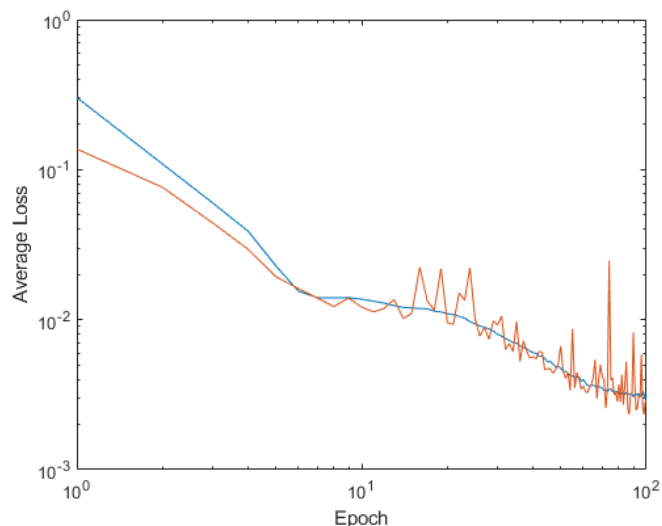
Erabilitako funtzio hau, maximo eta minimo ezberdinak ditu, [0,0] puntuan minimo absolutu bat izanik. Hurrengoko hau bere ekuazioa da:

$$z = -20 \cdot e^{\left(-0.2 \cdot \sqrt{\frac{1}{2}(x^2+y^2)}\right)} - e^{\left(\frac{1}{2}(\cos(2\pi x) + \cos(2\pi y))\right)} + 20 + e; x, y \in [-0.95, 1] \quad (14)$$

Non, x eta y sarean sarrerak dira eta z irteera. Irudia 59-n ikusi daitezke sarea entrenatu ostean, espero zan irteera ezkerrean, lortutako emaitzak erdian eta sortutako errorea eskuman. Aurreko esperimuntuan bezala, batezbesteko Lossaren bilakaera Irudia 60-n ikusi daiteke, berriz ere, kolore urdina ikasketarako erabili diren patroien Lossarekin erabili da eta laranja balidazio patroien Lossarekin. Kasu honetan, ikusi daiteke, balidazio Lossa zarata haundia pilatzen duela azkeneko entrenamenduko epokatan. Hau gerta daiteke, garrantzitsua den ezaugarri bat edo beste, balidazioko patroietan presentzia haundiagoa izatea, ikasketetakoetan baino eta, horregatik, sareak ikasketan zehar momentu baten gutxiesten badu, balidazioko errorea igo egiten da.

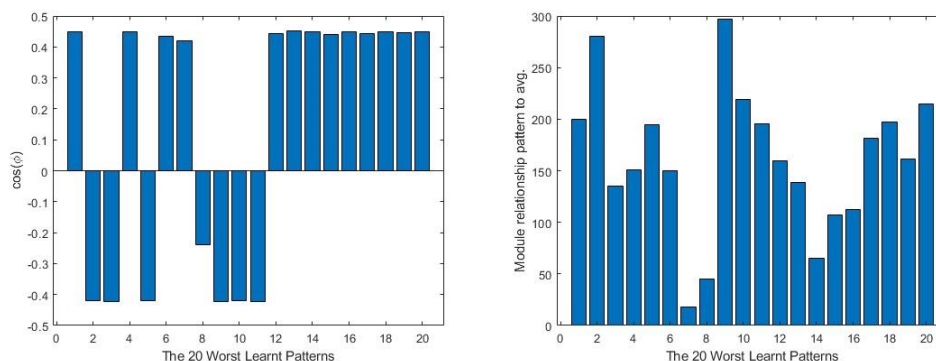


Irudia 59: Ackley funtzioaren entrenamendu emaitzak.

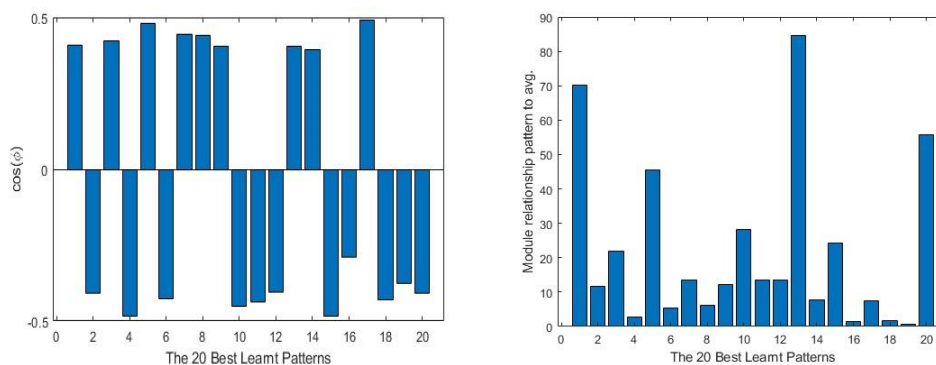


Irudia 60: Lossaren batezbestekoaren balioa epokaz-epoka.

Aurreko kasuan bezala, entrenatu ostean, patroien zuzenketak era bateratu baten atera dira, bakoitzaren modulu erlazioa batez bestearekiko eta haien arteko angeluaren kosinuarekin. Irudia 61-n ikusi daitezke txartoen ikasitako patroien emaitzak eta Irudia 62-n hoberen ikasitakoenak. Aurreko esperimientua bezala, txartoen ikasitako patroien moduluaren diferentzia askoz haundiagoa da, hoberen ikasitakoekin alderatura.

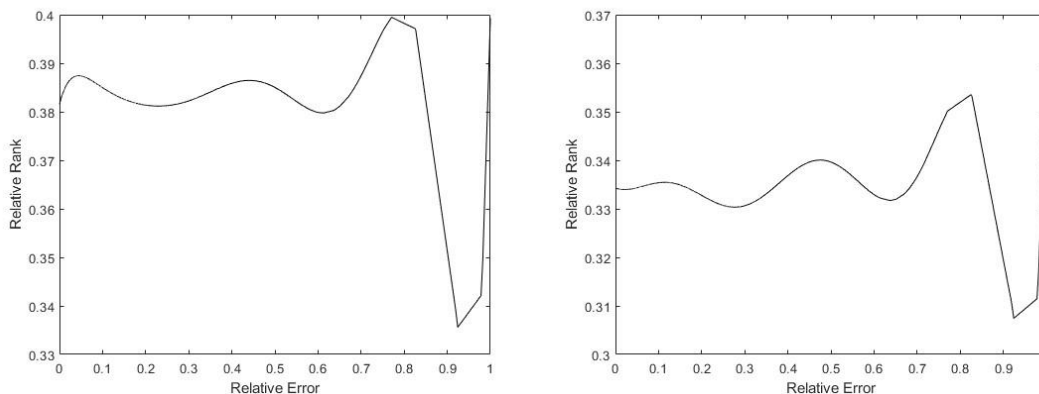


Irudia 61: Txartoen ikasitako patroien kosinua ezkerrean eta batezbestekoarekiko modulu erlazioa eskuman.



Irudia 62: Hoberen ikasitako patroien kosinua ezkerrean eta batezbestekoarekiko modulu erlazioa eskuman.

Prozedura geruzen errore erlatibo eta heina erlatiboaren arteko harremanaren partetik, lortzen den emaitza, aurreko esperimentuaren antzerakoa da, aipatzeko elementu bezala, errore maximoa inguruan dituen erroreek baino heina haundiagoa aurkezten duela. Irudia 63-n ikusi daitezke lortutako erregresio polinomikoen formak.



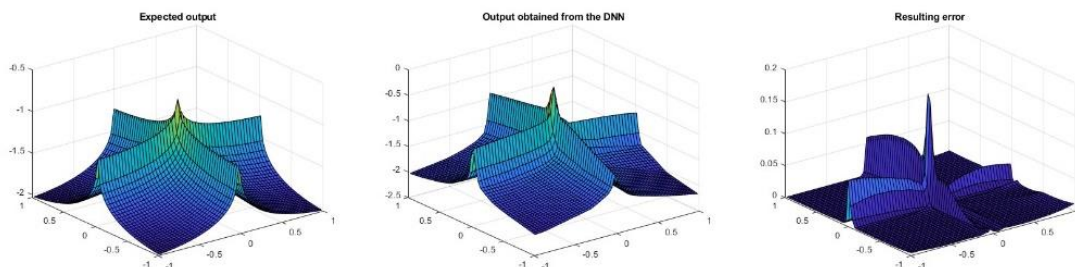
Irudia 63: Erregresio polinomikoa heina erlatiboa errore erlatiboarekiko. Ezkerrean X_2 egonda eta eskuman X_4 .

5.3. 3. Esperimentua, Cross-In-Tray funtzioa

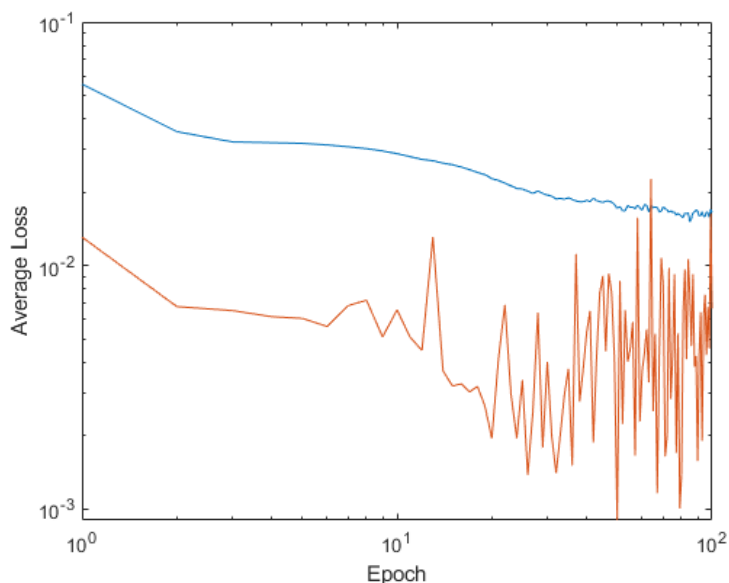
Funtzio hau, maximo lokal bat aurkezten du $[0,0]$ -n eta 4 minimo global, kasu honetako datuen mugan. erabilitako ekuazioa hurrengokoa hau da:

$$z = -0.0001 \left(\left| \sin(x) * \sin(y) * e^{\left| 100 - \frac{\sqrt{x^2 + y^2}}{\pi} \right|} + 1 \right| \right)^{0.1} ; x, y \in [-0.95, 1] \quad (15)$$

Non x eta y sarrerak diren eta z irteera. Irudia 64-n ikusi daitezke ezkerrean, nahi den sarearen irteera, erdian lortutako emaitza eta eskuman ateratako errorea. Aurreko kasuan bezala, Irudia 65-n sareak sortu duen Lossaren bilakaera ikusi daiteke, urdinez ikasketa patroiek sortzen dutena eta laranja balidaziokoena. Aurreko kasuekin konparaturik, batezbesteko Loss balorea altu xamar geratu da 100-garren ikasketa epoka eta gero. Errorearen grafika ikusita, funtzioa sortzen dituen ertzak ez dira ondo ikasten, funtzio honetan sare honekin.

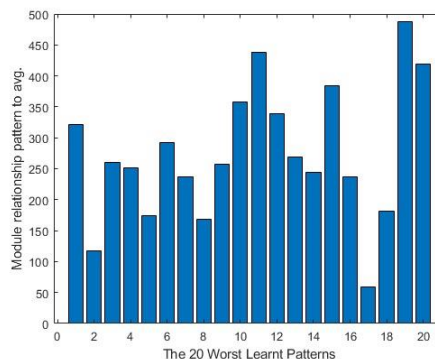
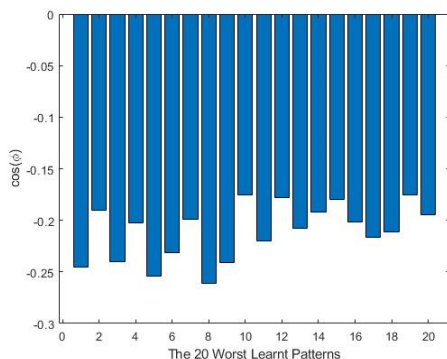


Irudia 64: Cross-In-Tray funtzioaren emaitzak.

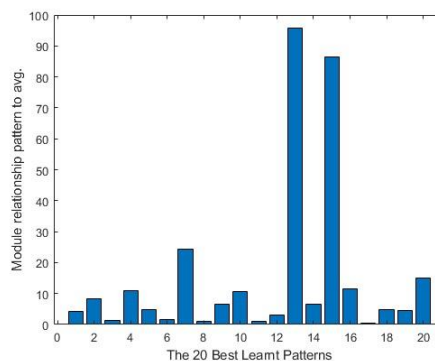
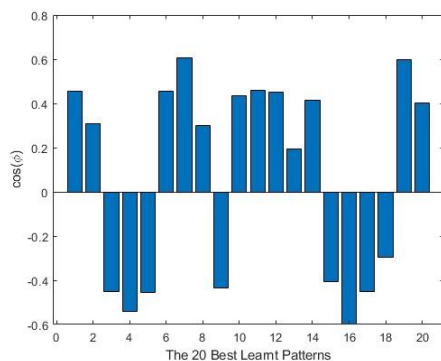


Irudia 65: Lossaren batezbestekoaren balioa epokaz-epoka

Patroien jokaera berdintsu izaten jarraitzen du, txartoen ikasten diren patroien modulu erlazioa batezbestekoarekiko, hoberen ikasten direnarekiko askoz altuagoa da. Kasu honetako aipamen berezi bezala, txartoen ikasten diren patroiz guztiek kontrako norantza aurkezten dute. Irudia 66 eta Irudia 67-n ikusi daitezke patroien emaitzak.



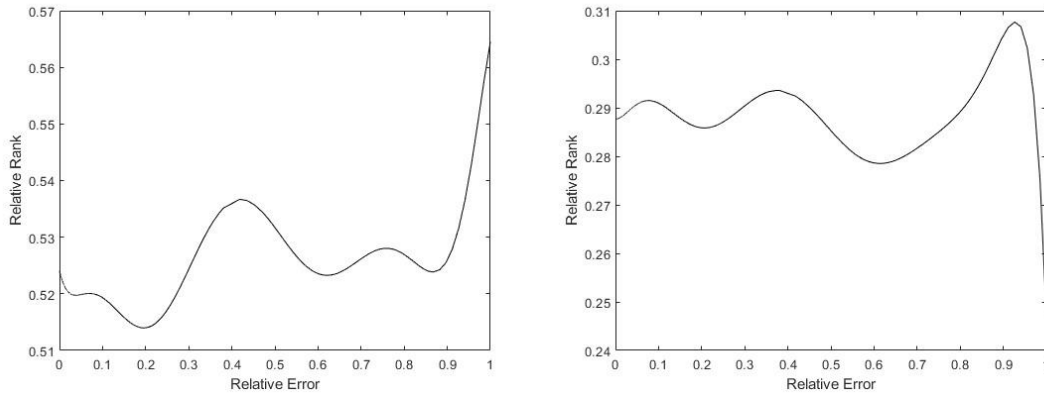
Irudia 66: Txartoen ikasi diren patroien zuzenketen angeluaren kosinua ezkerrean eta moduluen erlazioa eskuman.



Irudia 67: Hoberen ikasi diren patroien kosinu erlazioa ezkerrean eta moduluen erlazioa eskuman.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

Errore/eina erlazioaren aldetik, prozedura geruzetan, X_4 jokaera antzerakoa aurkezten du, aurreko esperimientuekiko, baina, X_2 ostera, heina handitzeko joera erakusten du errorea handitu ahala. Hau gerta daiteke, Loss funtzioaren gradientearen norantza patroien batezbestekoarekin lerrokatu gabe egoteagatik. Hau, Irudia 68-n ikusi daiteke.



Irudia 68: heina eta errore erlatiboaren arteko elazioaren erregresio polinomikoa. X_2 ezkerrean eta X_4 eskuman.

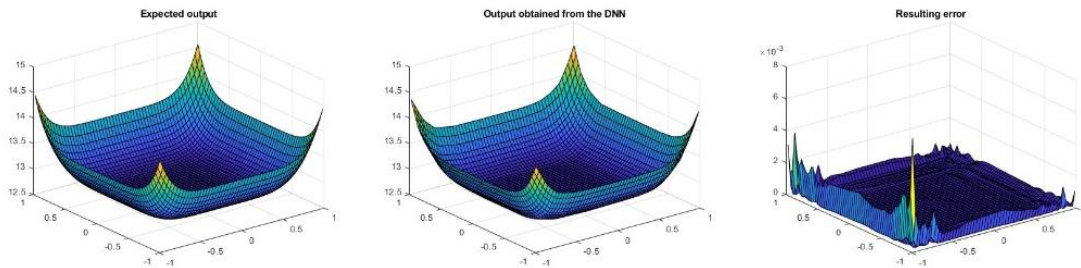
5.4. 4. Esperimentua, 5. De Jong funtzioa

Funtzio hau, erabilitako sarreraren eremurako, laua den minimo bat eskaintzen du, eta izkinatan igo egiten da. Hurrengoko ekuazioa jarraitzen du:

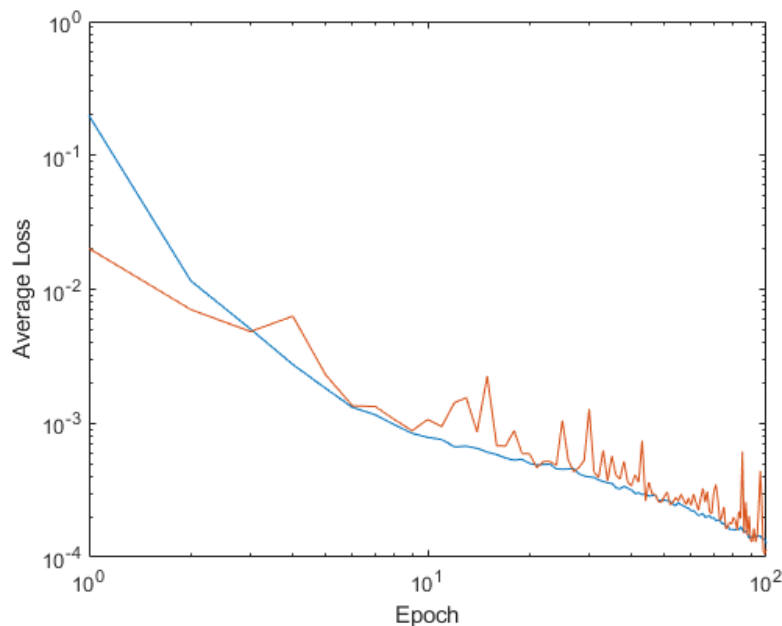
$$z = \left(0.002 + \sum_{i=1}^{25} \frac{1}{i + (x - a_{1i})^6 + (y - a_{2i})^6} \right); x, y \in [-0.95, 1]; \tag{16}$$

$$a = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 \dots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 \dots & 32 & 32 & 32 \end{pmatrix}$$

Non x eta y sarrerak diren eta z irteera. Irudia 69-n ikusi daitezke, esperotako irteera, lortutakoa eta ateratako errorea. Lehen bezala, Irudia 70-n sareak sortu duen Lossaren batezbesteko balioa epokaz, epoka aukitzen da, berriro ere urdina ikasketarako erabili diren patroien Lossa jasotzen du eta laranja, balidaziorako erabili direnena. Kasu honetan, sarea nahiko ondo ikasteko ahalmena du funtzio hau ikasteko.



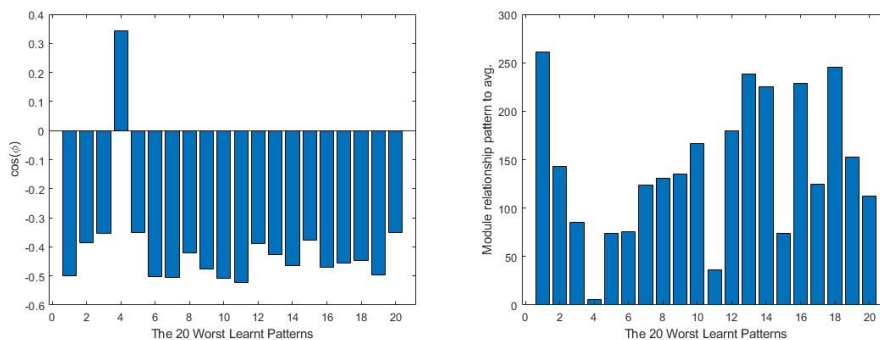
Irudia 69: De-Jong funtzioaren emaitzak.



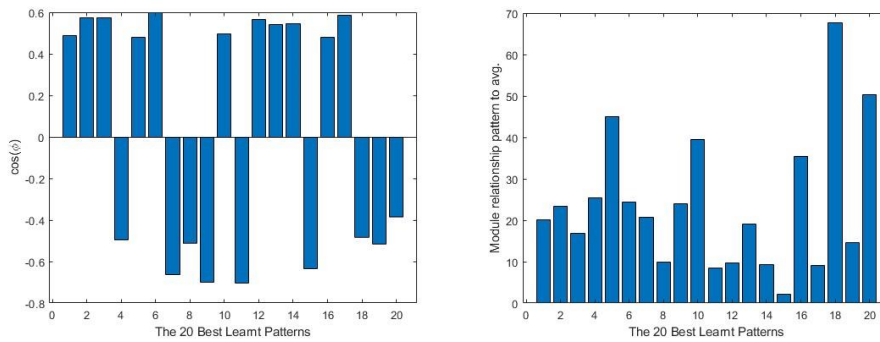
Irudia 70: Lossaren batezbestekoaren balioa epokaz-epoka.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

Patroien aldetik, aurreko esperimentuaren oso antzerakoa du, bi parteetan esperotako jokaera dauka. Irudia 71 eta Irudia 72-n ikusi daitezke jokaera hauek.

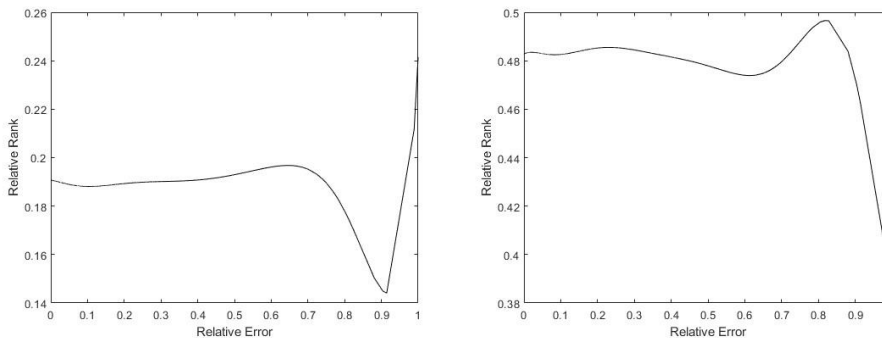


Irudia 71: Txartoen ikasi diren patroien kosinua ezkerrean eta moduluen erlazioa eskuman.



Irudia 72: Hoberen ikasi diren patroien kosinua ezkerrean eta moduluen erlazioa eskuman.

ReLU-etako Heina/errore erlazioan, aurreko kasuetan aurkitutako antzerako egoera suertatzen da. X_2 -n heinak joera positiboa du errorearekiko, eta aurreko kasuetan bezala, honek islatzen du, patroiren bat dagoela errore altua duena eta batezbestekoarekiko ez dagoela batere lerrokaturik. X_4 -n bere partez, lehenengo esperimentuan lortutako emaitzekin bat dator. Hau Irudia 73-n ikusi daiteke.



Irudia 73: X_2 -n ezkerrean eta X_4 -n eskuman sortzen den Heina/errore erlatiboko erregresio linealak.

V. Proposatutako aldaketak Gradiente Estokastiko Beherakorraren ikasketan

II-3 eta IV puntuetan esan den moduan, SGD algoritmoa oso erabilia den algoritmo bat da, batez ere oso erabilia guztiz konektaturiko geruzetan oinarritzen diren sare neuronaletan[72], [305], [323]–[326]. Hain erabilia izateko arrazioa, era aleatorio baten Loss minimoko zonaldera azkar iristeko ahalmena duelako da, beste algoritmo batzuekin konparaturik. Ala ere, bere bertuterik haundiena, bere ahultasunik haundiena da, ez bait da bera bakarrik, era natural baten geldirik geratzen Loss minimora heltzen denean, baizik eta, bere inguruan etengabe mugitzen jarraituko du[72].

IV-1 puntuan, deritzon ikasketa ratio bat aplikatzen dela azaldu egiten da. Ratio hau, azkeneko patroiaeren Lossari, pisu sinaptiko batekiko, aplikatu egiten zaio, portaera ezegonkorak ekiditeko. Errorea irteeratik sarrerara igarotzen denez gainbegiratutako ikasketan, geroz eta sakonago egon pisu sinaptiko bat irteeratik hasita, zuzenketa gutxiago jasango du ikasketa prozesuan. Fenomeno hau, Lossaren gradientearen desagerketagatik gertatzen da eta hau saretik igarotzen denean, eta iristen den errorearen zatia, sarearen inguru sakonetara ez da nahikoa eragin nabarmen bat sortarazteko. Honek, ikasketa epoka gehiago eskatuko ditu pisu sakon hauek mugitu ahal izateko. Orohar, sare neuronalaren ikasketa algoritmoaren parametroak jesartzen direnean finkatu egiten da eta momentu horretatik aurrera mantendu egingo da ikasketa prozedura guztian zehar. Balore hau oso haundia suertatzen bada, pisu sinaptikoak era ezegonkor baten modifikatuko dira ikasketan zehar, gradienteak filtro barik aplikatuko bait delako, batez ere irteeratik gertu dauden pisuetan, non Lossaren gradienteak ez du ia desagerpenik jasan. Bestalde, oso txikia bada, pisu sinaptikoak geldikor geratuko dira, gradienteak influentziarik ez baitu eraginik izango. Sarearen diseinatzailearen aurreko jakituria garrantzitsua da oso, balore zuzena ezartzeko; ikasketa prozedura hasieratik errepikatzeko arriskua dago, ondo definitzen ez bada.

Beste alde baterik, IV-1 puntuan ere azalduta, Dropout teknikaren bat erabili oi da, sarearen gehiegizko ikasketa ekiditeko [3],[58],[72],[319],[327],[328]. Esan den moduan, metodo hau, gradientearen bidez sortutako pisu sinaptiko batzuen zuzenketak deuseztatu egiten dira era aleatorio baten. Honekin pisu sinaptiko batzuen ikasketa ez da osoa suertatzen, helburu den funtzioaren ikasketa orokorrago bat lorturik. Ikasketa ratioarekin bezala, Dropout bidez ikasi gabe geratzen diren pisuen portzentuala oso altua bada, sareak ez du informazio nahiko jasoko patroietatik, ikasketa prozedurari ahalmena kenduz. Bestalde, oso baxua bada, gehiegizko ikasketa kasutik hurbilago egongo da sarea, patroien kopurua txikia bada edo epoka ugariko entrenamendu bat proposatzen bada. Aurreko kasuan bezala, sarearen diseinatzailearen esperientzia oso garrantzitsua da balore egokia emateko, ikasketa prozesua berriz errepikatzea beharrezkoa izatea ekiditeko.

1. Ikasketa ratioa eta Dropout ratioa: Aurkitutako problemak, analisia eta aurkitutako irtenbideak

Bi parametro hauetan oinarriturik, bi konponbide aurkitu dira. Lehenengoa, ikasketa ratioa, α , aldakorra bihurtzea da. Kasu honetan, sare neuronalaren posizioaren arabera ikasketa ratioa aldatu egiten da. Bigarrena, Dropout ratioa, ikasketa epoka bakoitzean aldatuko da. Bi aldaketa hauen helburua, Loss funtzioaren beherakada bat sortzea da, ikasketa prozesua behar duen denbora asko handitu gabe. Proposaketaren prozesuan, aldaketa bakoitza frogatu da bere kabuz eta loturik, ikasketa gertatzen den eragina aztertzeko.

1.1 Alpha dinamikoa

Hobekuntza hau, geruzaren arabera aldakorra den ikasketa baten deritzo. Bere funtsa, ikasketa epoka berdinerako ikasketa ratioa igo egiten du, pisu sinaptikoa sarreratik geroz eta hurbilago dagoenean. Ikerketaren ondorio, ikusi egin da SGD algoritmoaren entrenamenduan erroreak sarearen zehar egiten duen higidura, sareko sarreraren gertu dauden pisu hauen ikasketa atzeratu egiten duela, hauek mantsoago ikastera eramanez, irteeratik gertu dauden pisuekin alderaturik. (17)-(20) artean, ikusi daitezke erabilitako ikasketa ekuazioak lan honetarako.

$$w_{i,j,n}(t+1) = w_{i,j,n}(t) + \Delta w_{i,j,n}(t) \quad (17)$$

$$\Delta w_{i,j,n}(t) = \alpha \left(\frac{\partial Loss}{\partial w_{i,j,n}(t)} \right) + \beta \cdot m_{i,j,n}(t) \quad (18)$$

$$m_{i,j,n}(t+1) = m_{i,j,n}(t) + \Delta w_{i,j,n}(t) \quad (19)$$

$$\alpha = \alpha_0 \left(1 + \xi \frac{n}{N_{layers}} \right) \quad (20)$$

Non, $w_{i,j,n}(t)$ n garren geruzan i sarrera eta j irteera konektatzen dituen pisu sinaptikoa den, $m_{i,j,n}(t)$ pisu horretarako aplikatzen den momentua eta $\left(\frac{\partial Loss}{\partial w_{i,j,n}(t)} \right)$ pisu horretarako Loss funtzioaren deribatu partziala. ikasketa ratioa da, modifikatu egingo dena eta aplikatzen den momentuaren ratioa. (20)-garren ekuazioan proposatzen den aldaketaren funtsa dago, non baldintza aldakor bat proposatzen da. Baldintza hau, n , N_{layers} , ξ eta α_0 -ren menpe dago. α_0 ikasketa ratio basea da eta ξ , ikasketa ratioaren iguera sarean zehar islatzen du.

1.2 Dropout dinamikoa

Dropout teknika, oso ezaguna den teknika bat da. Honekin, $P_{dropout}$ probabilitatearekin geruzaz geruza eta epokaz epoka, pisu sinaptiko batzuen ikasketa 0 bilakatzen da. Normalean faktore hau entrenamendu guztian zehar mantendu egiten da, baina hurrengo ekuazioan planteatzen den aldaketa agertzen da.

$$P_{dropout} = P_{dropout,0} \cdot e^{-\frac{t}{T}} \quad (21)$$

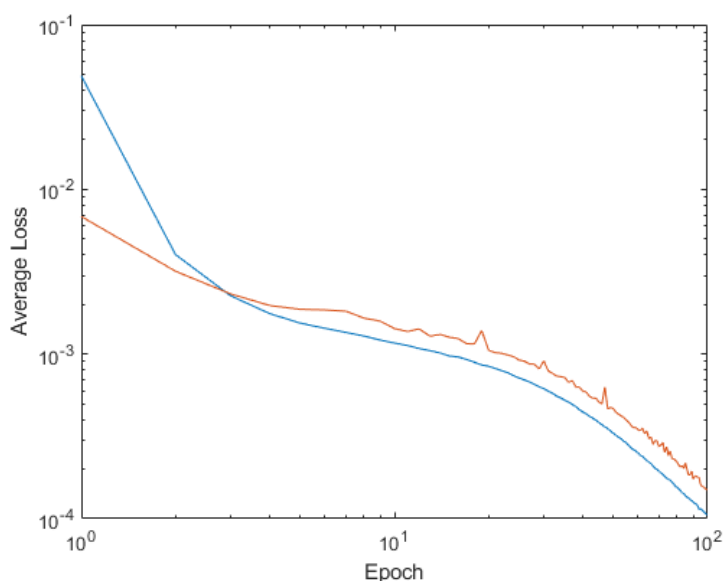
Non, $P_{dropout,0}$ hasierako probabilitatea den, t oraingo ikasketa epoka eta T desagerpen koefizientea. Era honetan, sarearen ikasketa hasieran, geruza bateko pisu sinaptikoak probabilitate berdina daukatenez, Dropout haundiena epoka honetan aplikatzen da, sarea era orokorrago batean ikasi dezan, detaile txikienak kontutan izan barik. Epokak joan ahala, Dropout hau jaitsi egiten da, sarea dagoeneko doituta baitago eta hasieran deuseztatu diren detaileak ikasi ditzake, emaitza orokorrak emateko kapazitatea galdu gabe.

2. Egindako esperimentuak eta beraien emaitzak

IV-5 puntuan erabilitako sare neuronal berdinetik hasita, ikasketa prozesuz berriz abiarazi da, datuak konparatzeko, aurreko kasua, alpha dinamikoa aplikaturik, Dropout dinamikoa eta, azkenik, bi aldaketak batera lan egiten. Aurreko kasuan bezala, lehendabizi mexikar kapela deritzon funtzio sinplea erabili da kasu guztiak frogatzeko eta, gainontzeko kasuetan, bakarrik entrenamendu normala eta bi aldaketak baturik egin da. Erabaki hau, lehenengo kasuen, bi tekniken konbinazioa, bakoitza bere aldetik aplikatu beharrean, eragin nabarmena lortzen delako izan da.

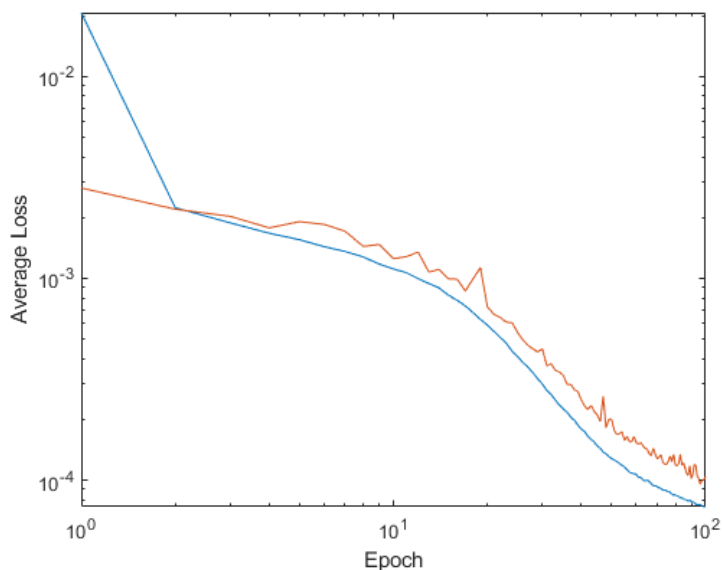
2.1. 1. Esperimentua, Mexikar kapela

Esperimentu honen funtzioaren ekuazioa (13) da, IV-5-n ikusi daitekeena. Entrenamenduaren ostean, 100 epoka igarota, lortu den ikasketa patroien errorea 0.00010586-en geratu da eta balidazio patroiena 0.000150754-n. Irudia 74-n ikusi daiteke batezbesteko Lossaren eboluzioa entrenamenduan zehar.



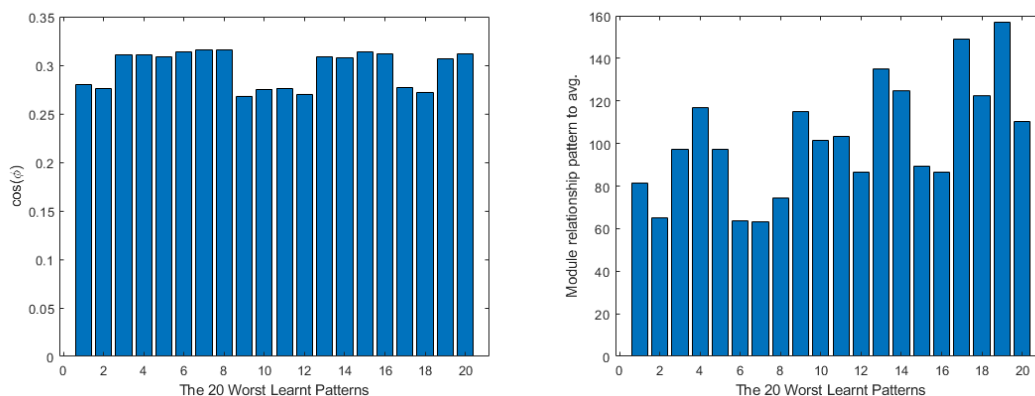
Irudia 74: Batezbesteko Loss funtzioaren balioa, urdina entrenamendu patroiena izanik eta laranja balidazio patroiena.

Alpha dinamikoa aplikatu ezkerre, lortzen den Loss-aren balioa 0.000073996-ekoa da entrenamendu patroietan eta 0.000104274 balidaziokoetan. Irudia 75-n ikusi daiteke zein izan den Loss honen eboluzioa.



Irudia 75: Batezbesteko Loss funtzioaren balioa alpha dinamikoa aplikatuta, urdina entrenamendu patroiena izanik eta laranja balidazio patroiena.

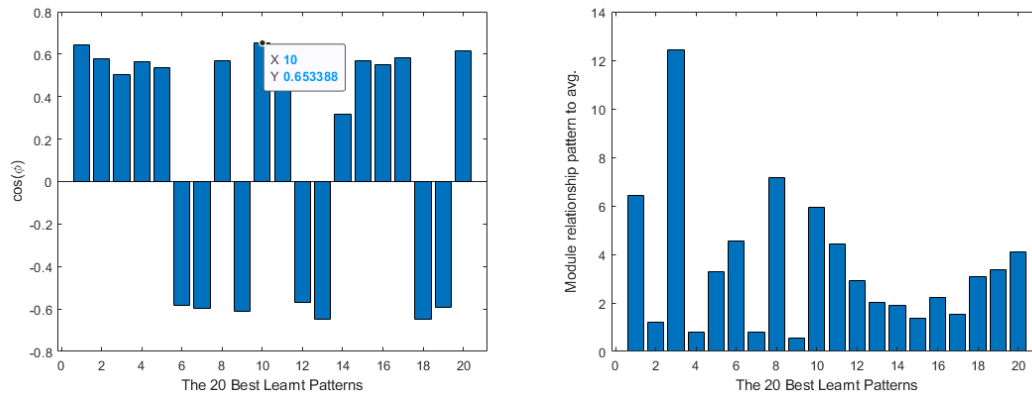
Zuzenketa bektoreen biderketa eskalarra aplikatuta, alpha dinamikodun ikasketa berri honetan, eta IV-5-en bezala ordenatu dira, 20 txarren ikasi diren patroien baloreak eta 20 hoberen ikasi direnenak atera egin dira. Irudia 76-n txartoen ikasi diren patroiek ikusi daitezke. Aurreko kasuarekin konparaturik, beraien kosinua pixkat igo egin da, eta moduluen erlazioa asko txikitu da. Honek esan nahi du, txartoen ikasi diren patroiak, bestetikiko lerrokatu direla



Irudia 76: Alpha dinamikoa aplikatuta, txartoen ikasi diren patroien $\cos(\phi)$ ezkerrean eta moduluen erlazioa eskuman.

Bestalde, Irudia 77-n, hoberen ikasi diren patroien datuak ikusi daitezke. Lortzen diren kosinuen balioak 0.8-tik gertu gertzen dira era absolutuan eta modeloen erlazioa 1:1 oso gertu geratzen da, txarren ikasten dinarekiko.

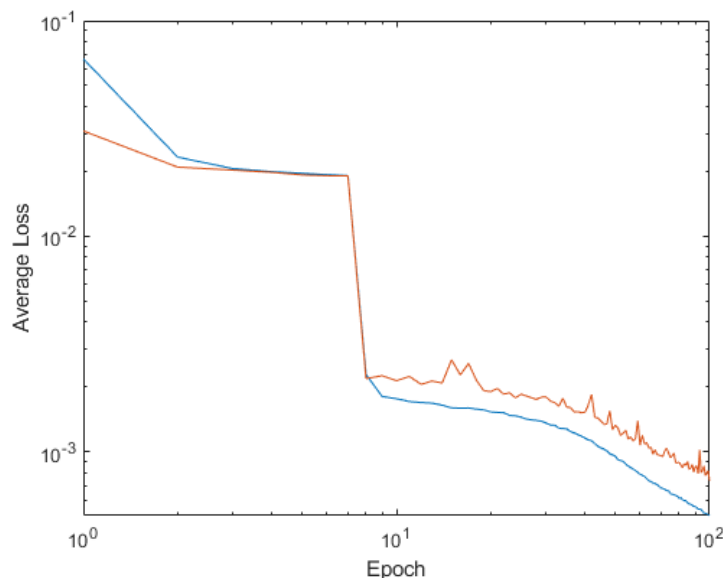
Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak



Irudia 77: Alpha dinamikoa aplikatuta, hoberen ikasi diren patroien $\cos(\varphi)$ ezkerrean eta moduluen erlazioa eskuman.

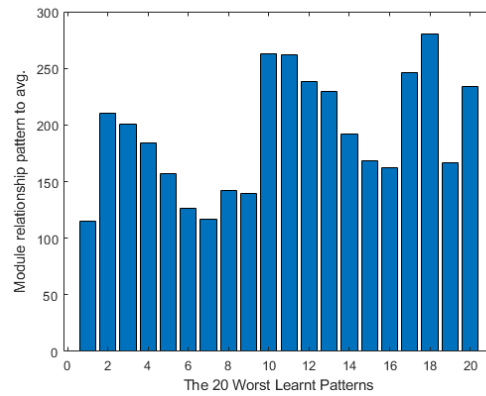
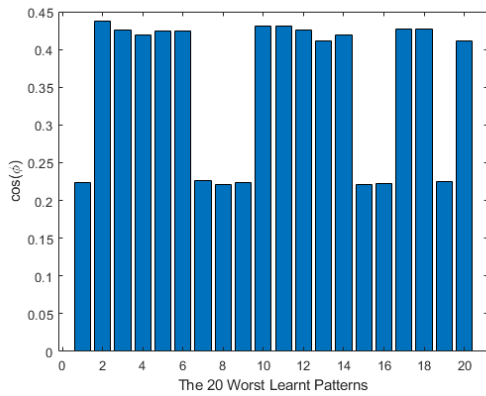
Orohar, esan daiteke aldaketa honek SGD algoritmoa hobetzen duela kasu honetarako. Hoberen ikasten diren patroiak oso lerrotuta geratzen dira batezbestekoarekiko eta txartoen ikasi direnak, bestalde, oso influentzia txikia daukate ikasketetan.

Dropout-aren aldaketa aplikaturik egindako entrenamenduan, bere partez, ikasketa patroietan lortu den Lossaren balioa 0.000248867, originala baino bi aldiz haundiagoa eta 0.000330196 balidazio patroietan. Zenbaki ez onuratsuak izan arren, esperimentuarekin jarraitzea erabaki egin zen. Irudia 78-n ikusi daiteke Loss honen bilakaera.

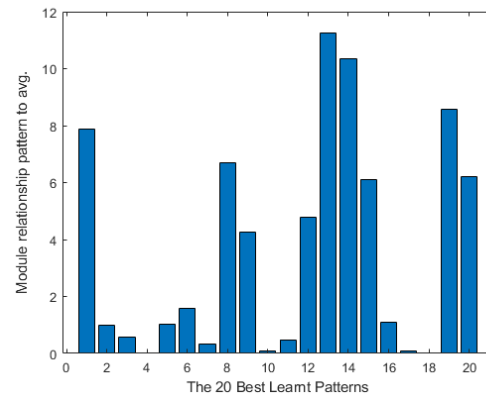
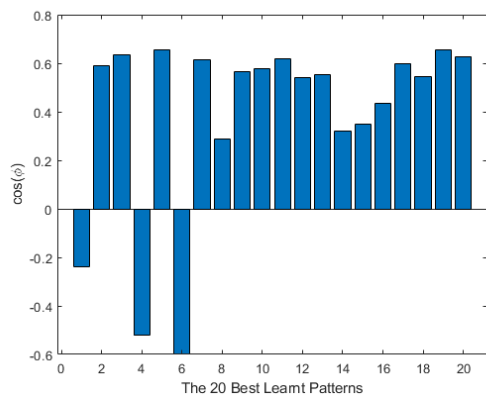


Irudia 78: Batezbesteko Loss-aren balioa entrenamendu patroietan urdinez eta balidaziokoetan laranja.

Patroien ikasketaren aldetik, bai txartoen ikasi diren patroiak, bai hoberen ikasi direnak, antzerako portaera izan dute alpha dinamikodun aldaketarekin alderatuta. Irudia 79-n txartoen ikasi diren patroien datuak ikusi daitezke eta Irudia 80-n hoberen ikasi direnenak.

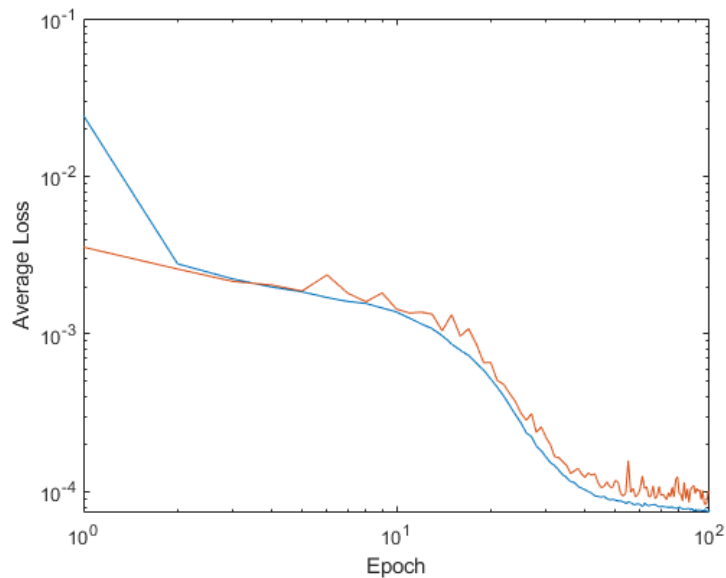


Irudia 79: Txartoen ikasi diren patroien datuak, ezkerrean kosinua eta eskuman moduluen erlazioa.



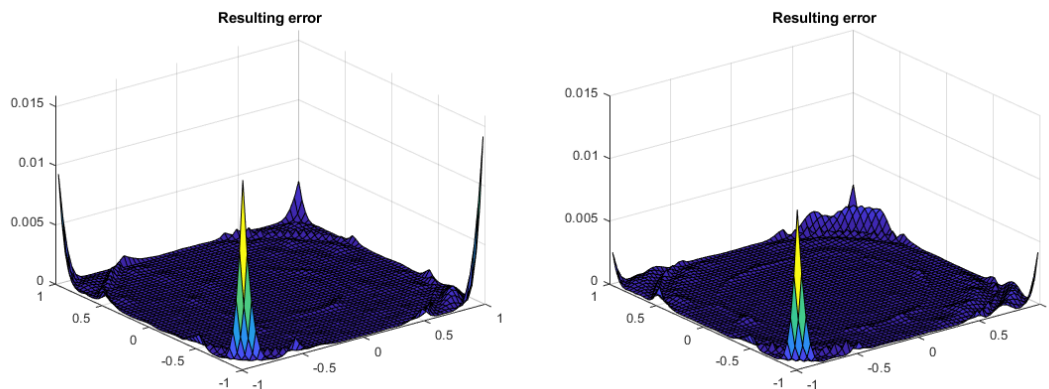
Irudia 80: Hoberen ikasi diren patroien datuak, ezkerrean kosinua eta eskuman moduluen erlazioa.

Azkenik, bi aldaketak batera aplikatu dira. Kasu honetan lortutako ikasketa patroien Loss-a 0.000076558-ra iritsi da eta balidazio patroiena 0.000099538-ra. Zenbaki hauek alpha dinamikoa bakarrik aplikatuz lortzen direnen oso gertu dago, ala ere, kasu honetan balidazio Loss-a entrenamendu patroienarekiko gertuago geratu da, gehiegizko ikasketaren urrunago geraturik. Irudia 81-n ikusi daiteke Loss-aren eboluzioa.



Irudia 81: Batezbesteko Lossaren balioa, usdinez entrenamenduko eta laranjaz balidaziokoa.

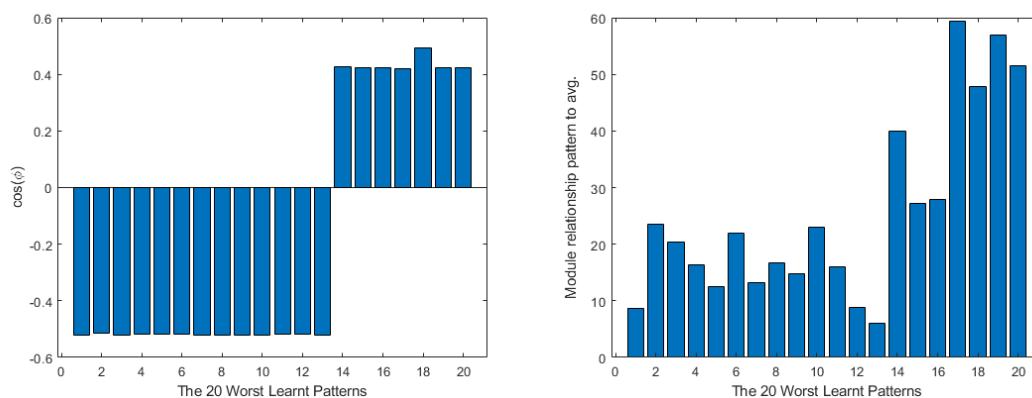
Baloreak ikusirik, pentsa daiteke konbinaketa ez duela zentsurik alpha dinamikodun soilik duen entrenamendua erabili daitekeela. Irudia 82-n ikusi daiteke nola bi aldaketak aplikaturik errorerik altuena asko jaisten dela, alpha dinamiko soilik aplikaturik lortzen denarekin konparaturik.



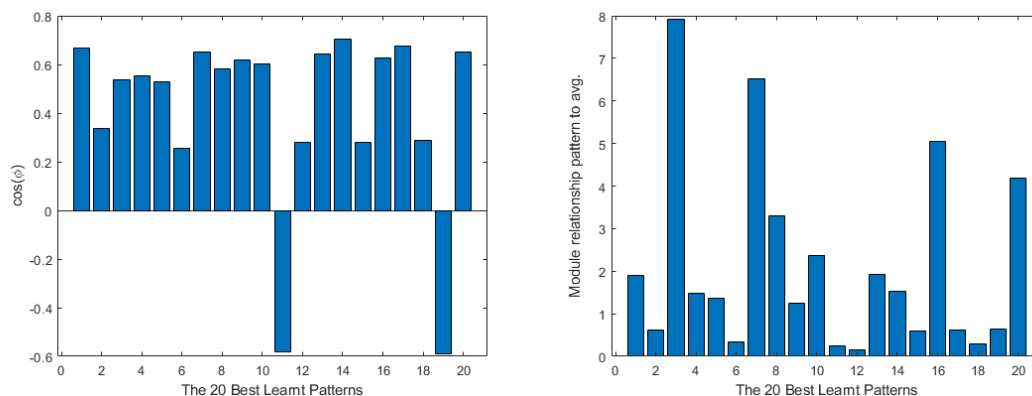
Irudia 82: Ezkerrean alpha dinamiko bakarrik aplikatuta lortzen diren erroreak patroiekiko eta eskuman bi aldaketak konbinaturik lortzen direnak.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

Patroien ikasketa ikerturik, ikusi daiteke nola txartoen ikasi diren patroiak batezbestekoarekin azkoz ere lerrokatuagoak geratu egin direla eta beraien modulu erlazioa asko jaitsi dela. Irudia 83- n ikusi daiteke lortutako baloreak. Hoberen ikasi diren patroiak, bere partetik, aurreko kasuetan lortutako antzerako baloreak lortzen dituzte, Irudia 84-n ikusi daiteke. Datu hauek mahai gainean edukita, esan daiteke bi aldaketak batera aplikatzea, onuratsua dela ikasketarako eta SGD algoritmo estandarra baino emaitza hobekoak lortzen dituela, batez ere txartoen ikasten diren patroien garrantzia igo egiten duelako batipat.



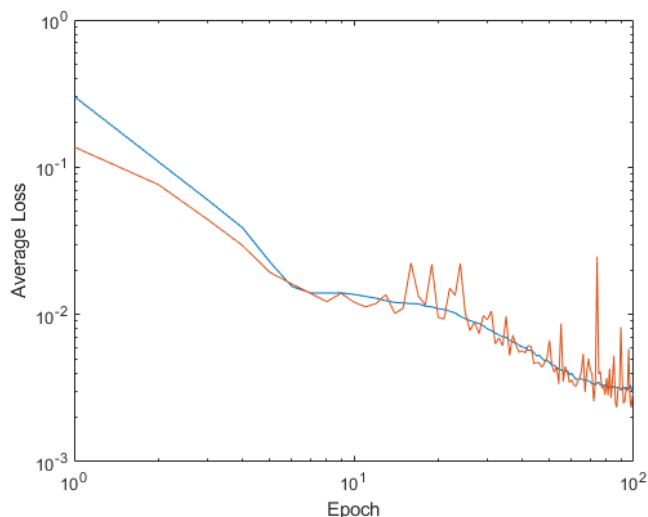
Irudia 83: Bi aldaketak batera aplikaturik, txartoen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.



Irudia 84: : Bi aldaketak batera aplikaturik, hoberen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.

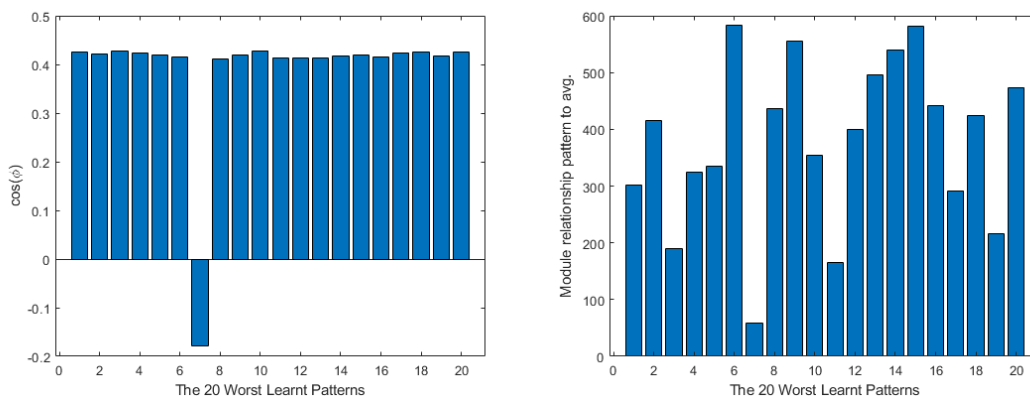
2.2. 2. Esperimentua, Ackley funtzioa

Ackley funtzioa IV-5.2-n erabilitako berdina da. [0,0] puntuan minimo absolutu bat aurkezten du eta honen inguruan maximo eta minimo lokal ugari. SGD estandarrekin entrenatu ostean, 0.003114478 Loss-a aurkezten du entrenamendu patroietan eta 0.002884209 balidaziokoetan. Irudia 85-n ikusi daiteke nola balidazio Loss-aren balioa oso zaratatsua suertatzen den azkeneko epokatan.



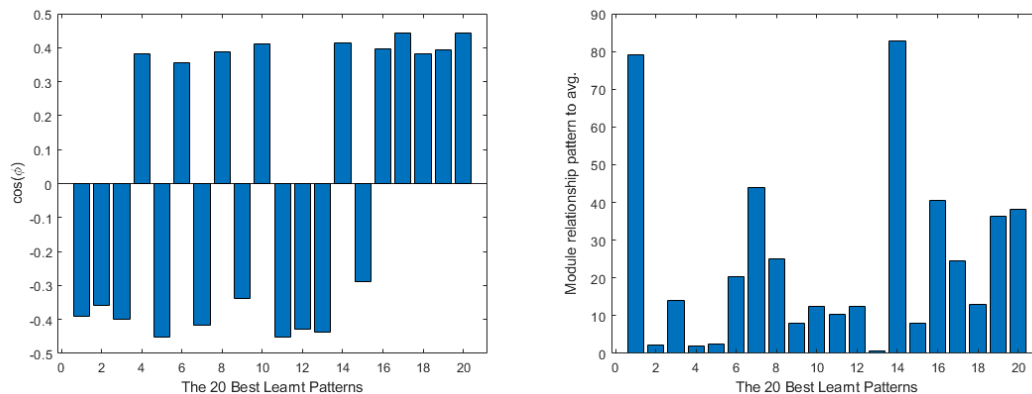
Irudia 85: Batezbesteko Loss-a Entrenamendu patroietan urdinez eta balidaziokoetan laranja.

Patroien ikasketetatik, aurreko esperimentuan aratako balore antzerakoak lortzen dira, bestalde normala den moduan. Irudia 86 eta Irudia 87, ikusi daitezke emaitza hauek.



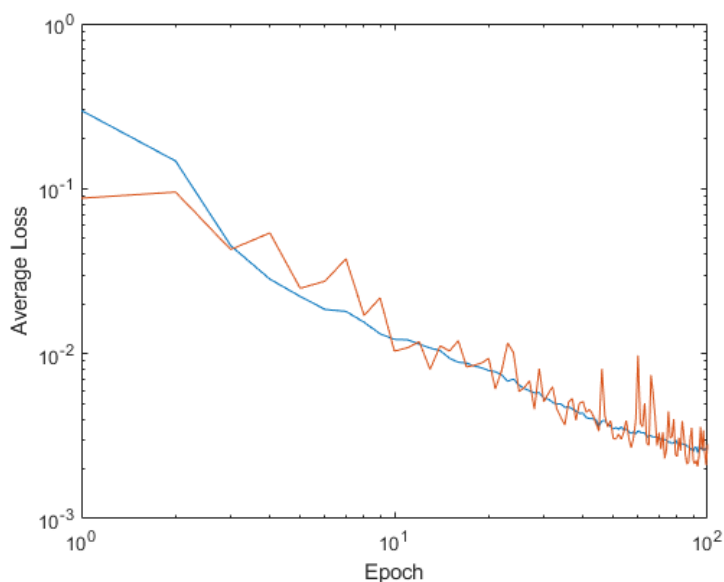
Irudia 86: Txartoen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak



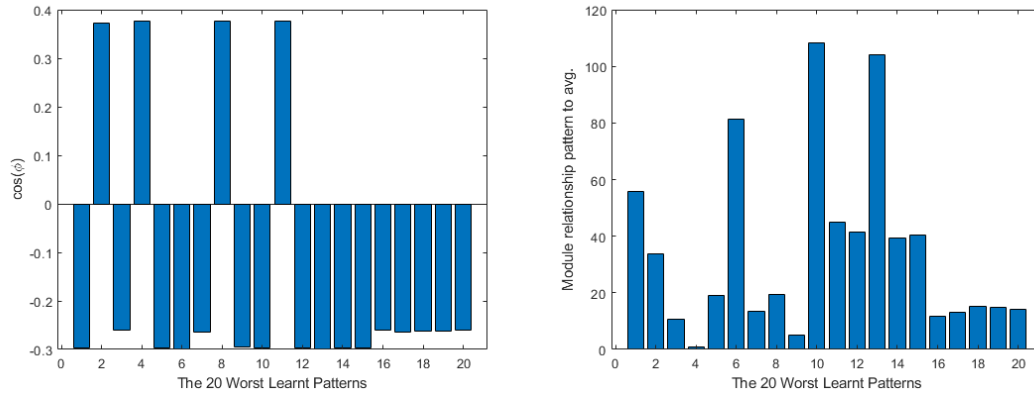
Irudia 87: Hoberen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.

Aldaketak aplikatu ostean, Loss-aren baloreak 0.00250811 entrenamendu patroietan eta 0.002813870 balidaziokoetan geratu dira. Irudia 88-n ikusi daiteke nola balidazio Loss-a zaratatsu izaten jarraitzen duela, baino aurreko kasuan baino gutxiago.

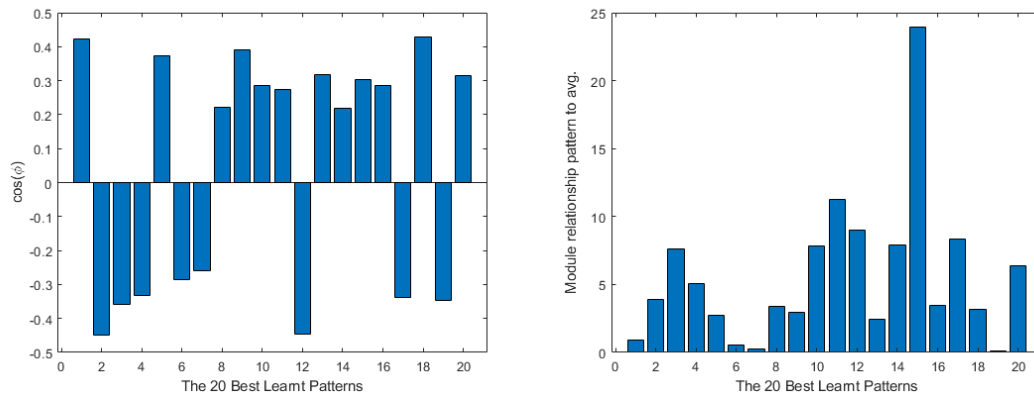


Irudia 88: Batezbesteko Loss-a Entrenamendu patroietan urdinez eta balidaziokoetan laranja.

Patroiak ikertuta, ez dira egon aldaketa haundirik egon ez txartoen ikasi diren patroietan, ez hoberen ikasi direnetan. Irudia 89 eta Irudia 90 ikusi daitezke emaitza hauek. Puntu onuratsu bezala, esan daiteke, moduluen erlazioa gertuago geratu da batezbestekoarekiko.



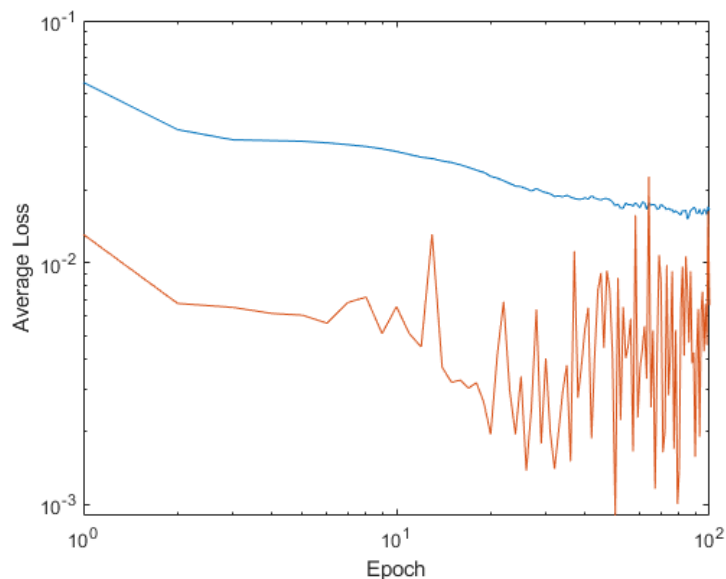
Irudia 89: Txartoen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.



Irudia 90: Hoberen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.

2.3 3. Esperimentua, Cross-In-Tray funtzioa

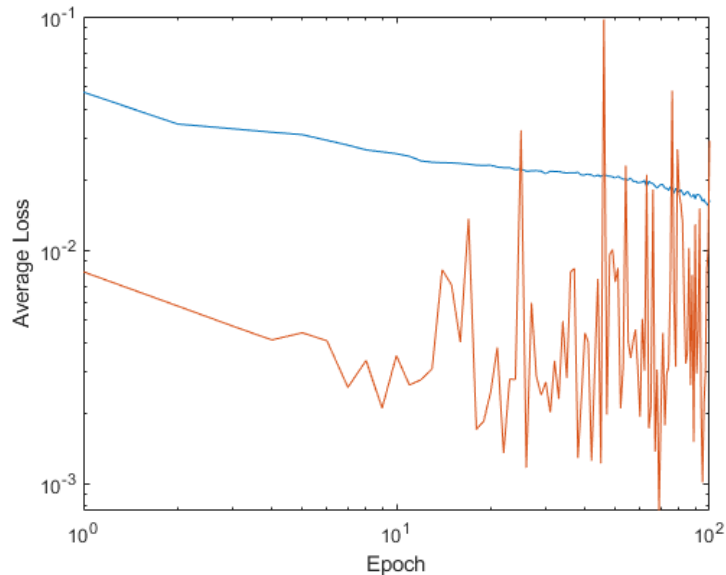
Esperimentu honen ekuazioa IV-5.3 puntuan egindakoaren berdina da. SGD estandararekin entrenatu ostean, entrenamendu patroien Loss balioa 0.016488167-en geratu egin da eta balidazioarena 0.006663838-n. Irudia 91-n ikusi daiteke hauen garapena.



Irudia 91: Batezbesteko Loss-a Entrenamendu patroietan urdinez eta balidaziokoetan laranja.

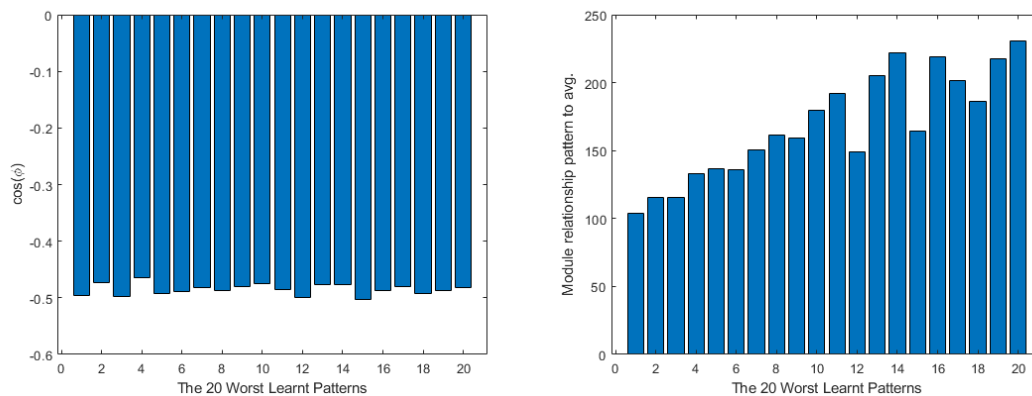
Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

Bi aldaketak aplikatu ostean, Loss-aren balioa 0.016340295 geratu da entrenamendu patroietarako eta 0.029352279 balidaziokoetan. Irudia 92-n ikusi daiteke bere garapena. Orohar ez da aldaketa nabarmenik sumatzen Loss-aren aldetik.



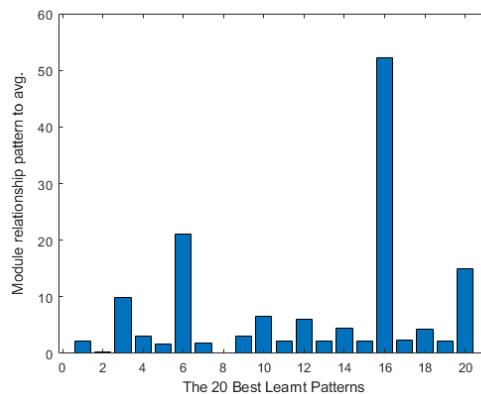
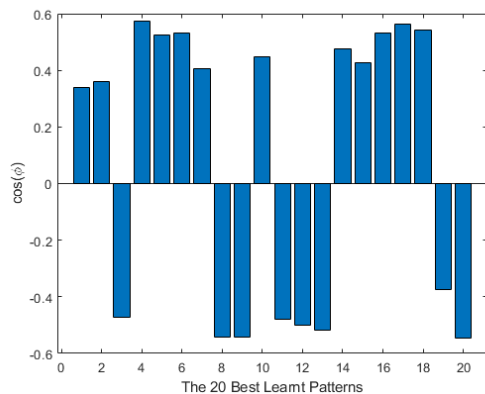
Irudia 92: Batezbesteko Loss-a Entrenamendu patroietan urdinez eta balidaziokoetan laranja.

Patroien entrenamenduaren aldetik, SGD normalaren entrenamenduak, IV ikusitakoaren antzera geratu egin dira. Irudia 93 eta Irudia 94-n ikusi daitezke emaitza hauek.



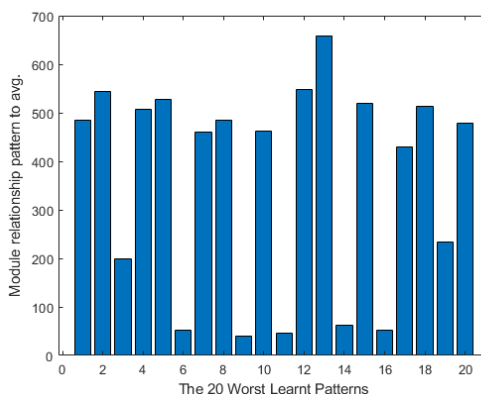
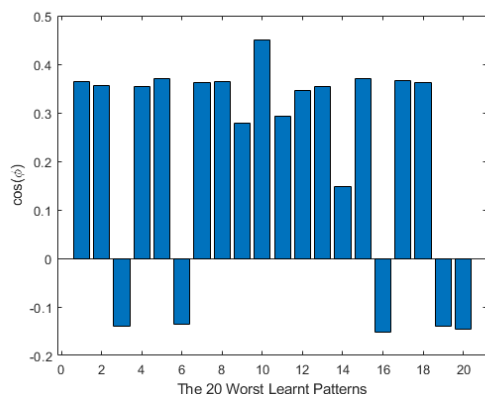
Irudia 93: Txartoen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

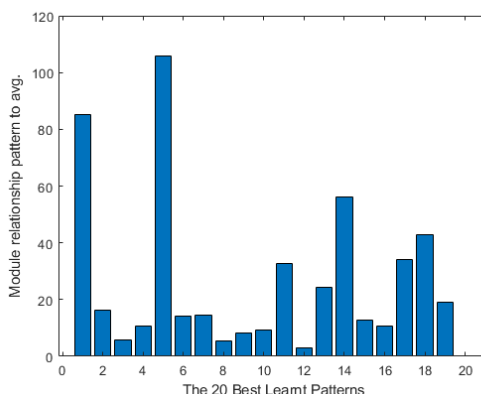
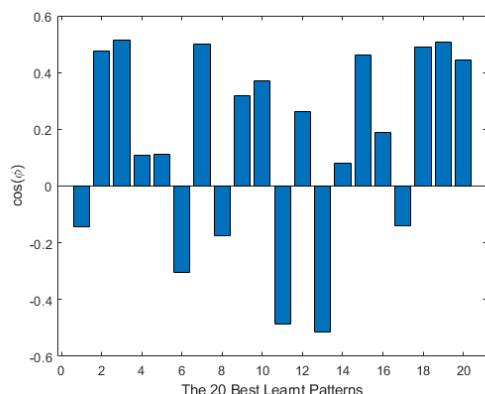


Irudia 94: Hoberen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.

Aldaketak egin ostean, Irudia 95 eta Irudia 96-n ikusi daitekeenez, patroien entrenamenduaren emaitzak gaiztotu egin dira. Parte ona begiraturik, txartoen ikasi diren patroien kosinu gehienak lerrokatu egin dira batezbestekoarekiko, baina beraien balioa txikiagotu egin da. Moduluen aldetik, kasu bietan hauek handitu egin dira batezbestekoarekiko.



Irudia 95: Txartoen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.



Irudia 96: Hoberen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.

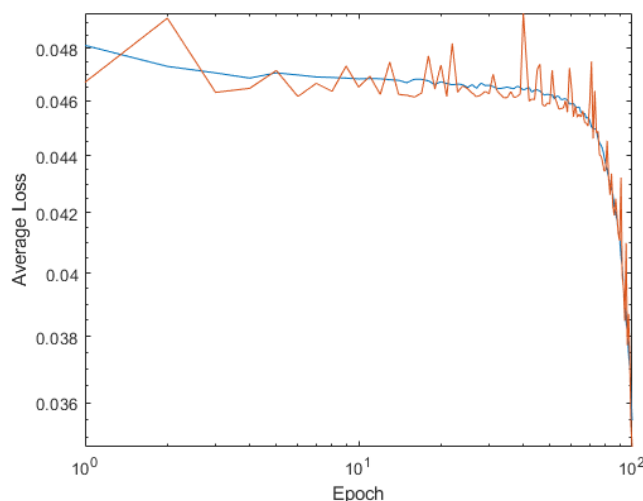
Orohar, esan daiteke aldaketek ez dutela efektu on bat izan adibide honetarako, ala ere, hasierako SGD-aren emaitzak ez ziren oso onak. Posible da, ekuazio hau ondo ikasteko sistemak epoka gehiago behar izatea edo sarea konplexuagoa bilakatzea.

2.4 4. Esperimentua Dropwave funtzioa

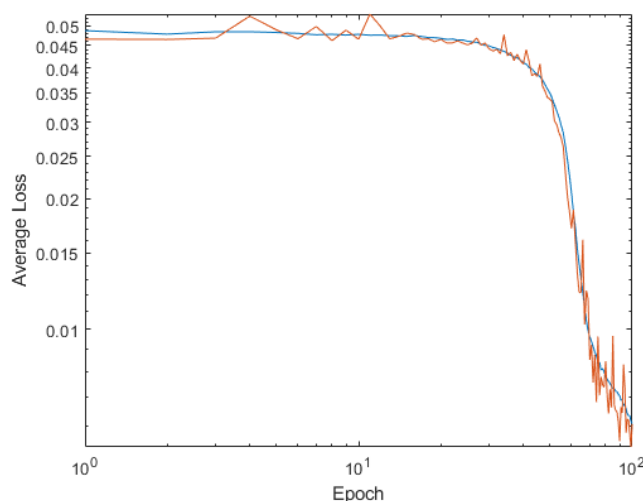
Esperimentu honen ekuazio helburua, hurrengoko hau da:

$$z = -\frac{1 + \cos(12\sqrt{x^2 + y^2})}{0.5(x^2 + y^2) + 2} \quad x, y \in [-0.95, 1] \quad (22)$$

Non z irteera den eta x eta y sarrerak. Aurreko kasuetan bezala 100 epokaz entrenatu egin da SGD normala aplikaturik eta gero zerotik, hasierako baldintza berdinetan, proposatutako bi aldaketak gehituz. Lehengo kasuan, 0.035496386-eko Loss bat lortu da entrenamendu patroietan eta 0.034738452-ekoa balidaziokoetan. Irudia 97 aztertuta, ikusi daiteke Loss-aren eboluzioa batez ere azkeneko epokatan sortu egin dela. Aldaketak aplikatu ostean, Loss-a 0.0066686 entrenatutako patroietan eta 0.006050894 balidaziokoetan. Loss-a 5 aldiz txikiagotu eginda. Ere, Irudia 98-n ikusten denez, 50 epokan hasi da Lossaren benetako eboluzioa, beste kasuan baino nahiko lehenago.



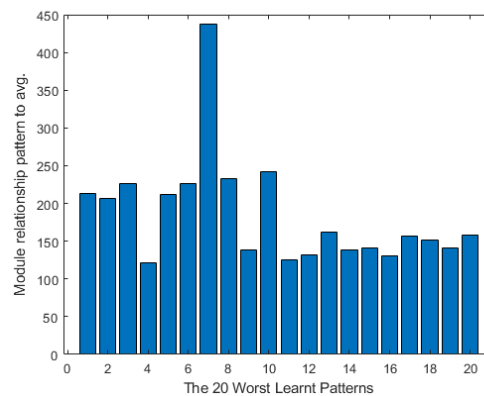
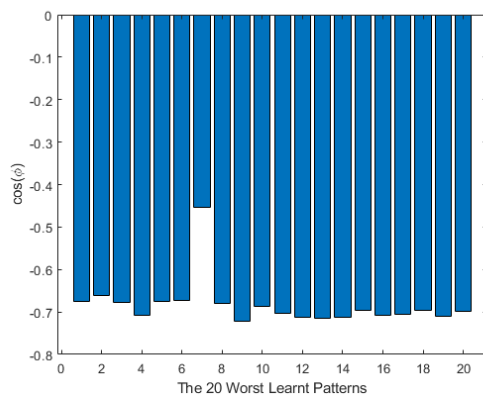
Irudia 97: SGD normalarekin batezbesteko Loss-a Entrenamendu patroietan urdinez eta balidaziokoetan laranja.



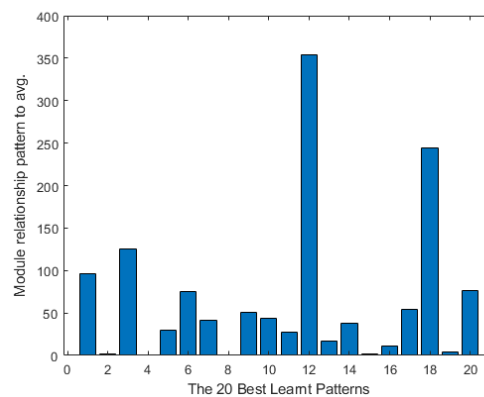
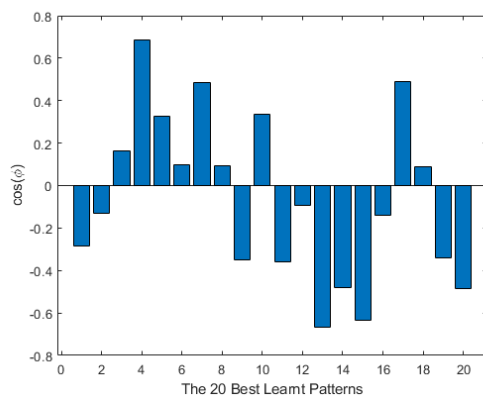
Irudia 98: : Aldaketak egin ostean batezbesteko Loss-a Entrenamendu patroietan urdinez eta balidaziokoetan laranja.

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

Patroien datuak begiratuta, SGD normalaz entrenatu diren patroien artean, Irudia 99-n ikusi daiteke nola txartoen ikasi diren patroiak guztiak kosinu negatibo bat aurkezten dutela eta beraien modulu erlazioa beti 100tik gora dagoela. Honek esan nahi du patroiak guztiak, ikasketa era kontrako baten mugitzen ahalegintzen ari direla. Bestalde, Irudia 100-n ikusi daiteke nola hoberen ikasi diren patroiak, batezbestekoarekin gutxi gorabehera lerrokaturik daudela baina beraien modulu erlazioa ez da oso esperantzarria. Orohar esan daiteke SGD entrenamendu honekin, sarea ez da kapaz ekuazio hau ikasteko.



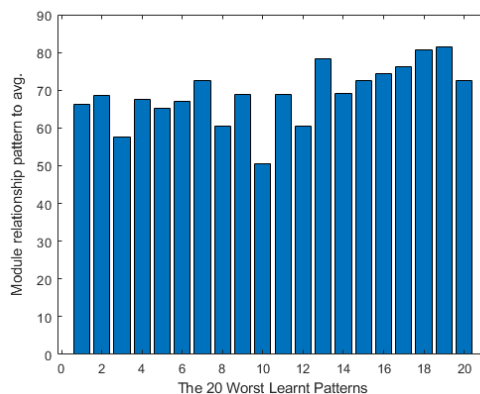
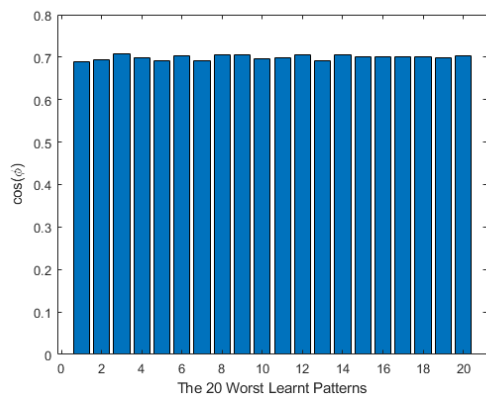
Irudia 99: Txartoen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.



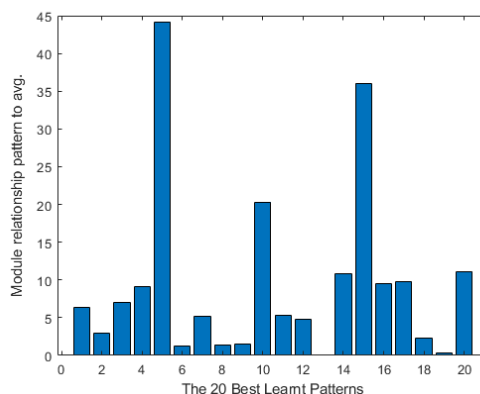
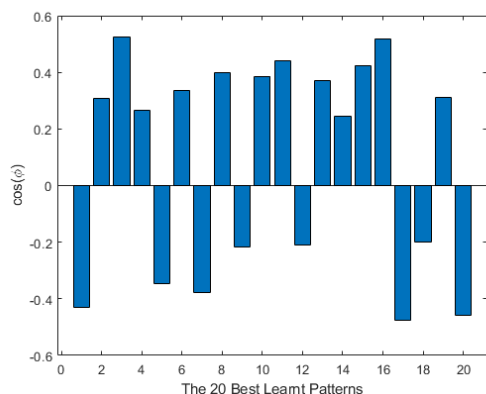
Irudia 100: Hoberen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

Bestalde, prestatutako aldaketak aplikaturik egoera guztiz aldatzen da. Irudia 101-n ikusi daitekenez, txartoen ikasi diren patroiak batezbestekoarekin oso lerrokatuta daude eta beraien moduluen erlazioa asko jaitsi da, aurreko kasuarekin konparaturik. Irudia 102-n hoberen ikasi diren patroiak ikusita, kosinuaren aldetik antzerako portaera aurkezten dute, baiian moduluen aldetik, askoz gertuago geratzen dira batezbestekotik. Azken finean, esan daiteke, SGD normalaz ikas ezin zen ekuazio bat, erabakitako sare neuronal konfiguraketarekin, ikasgarri izatera pasa dela aplikatutako aldaketekin..



Irudia 101: : Txartoen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.



Irudia 102: Hoberen ikasi diren patroien datuak, kosinua ezkerrean eta moduluen erlazioa eskuman.

VI. Ekarpenak, Ondorioak eta Hurrengo pausuak

Tesi honen garapenean, emaitza ezberdinak lortu egin dira. Alde batetik, sare neuronalak sortzeko liburutegia garatu egin da, beste liburutegi mota batzuek eskaintzen ez duten ikerketa datuak eskaintzen dituen eta sarearen konposaketa bera aldatzeko ahalmena ematen duena. Bestetik, liburutegi hau dela medio, sare neuronalen ikasketa ikertzeko aukera eman da, batez ere patroiek duten efektua ikusita; lortutako emaitzekin artikulu bat kaleratzeko adina datu eta konklusio zeudela ondorioztatu zen.

Momentu horretatik aurrera, SGD algoritmoa hobetzeko aukera zegoela ikusi zen eta liburutegian bertan aldaketak planteatu ziren hauek lortzeko. Aldaketak lortuta, berriz ere ikasketaren ikerkuntza martxan jarri zen, aurreko kasuarekin konparatuz. Lortutako emaitzak pozgarriak zirela ikusirik, bigarren artikulu bat kaleratzeko erabakia hartu zen. Bukatzeko, artikuluek aparte, seminario baten doktoretzako lana aurkeztea erabaki egin zen. Hurrengo puntuetan sakonago landuko dira aipatutako puntuak.

1. BasqueNet liburutegia

BasqueNet oso era pozgarrian lan egin dezakeen liburutegia bihurtu egin da. Era erraz batean sortu daitezke ikertu nahi diren sareak eta datu nahiko sortu ditzake ondorio onak ateratzeko, lehen komentatu den moduan hasiera baten azkeneko bertsioa baino datu gehiago gordetzeko diseinatu egin zen, baina eskura zegoen konputazio ahalmena kontutan harturik, azkeneko epokako datuak bakarrik gordetzea adostu zen; momentu honetan esperimentu bakoitzeko datu guztiak gordetzea 120MB inguru eskatzen ditu, epoka guztien datuak gorde ezker 12GB inguruko datu kopurua sortu egiten da, dena RAM memorian batera gordeta egon beharrezkin hauekin lan egin ahal izateko. Aldi berean, malgutasun nahikoa eskaintzen du beste ikasketa algoritmo batzuk sartzeko edota geruza mota gehiago. Orain dagoen puntura heltzeko ordu ugariko lana egin behar izan da, hasieran oso funtzio sinpleak garatuz eta gutxinaka sare bat betetzeko behar ziren elementuak sortzen joan da.

Lehenengo pausuetan, bi dimentsiotako funtzioekin lantzen hasi zen, prozedura guztiak landu ahal izateko, bakarrik sareak funtzio hauek ikasteko ahalmena lortu zuenean, hiru dimentsioetako funtzioekin lan egitera pasa zen. Azkeneko faseetan prozesua nahiko geldoa izan da, azkeneko funtzionalitateak sortutako sareak entrenatzen bukatzerakoan bakarrik aztertu ahal ziren eta gaizki joan ezker, kodea zuzendu eta gero berriro sare dena 0tik definitu beharra zegoen. Esperimentuetan erabili diren funtzioak hautatu den sarean entrenatzeko 20minutu inguru eskatzen dute, oso fin lan egiteko premia ezarrita sarearen garapenaren hasieratik.

Parte ez hain ona begiraturik, ez da lortu sarearen entrenamendua eta lan prozedurak txartel grafikoko GPUan exekutatzeko, baizik eta ordenagailuko prozesadore arruntean egiten du, prozeduren denborak luzatuz. Hau etorkizunerako lan ildo bat izan daiteke.

2. Ikerkuntzako emaitza

Orohar, ordu askotako simulazioak eta ideien birplanteatzeak ekarri dituzte. Azkenean, patroien ikasketaren prozesua hobeto ulertzea lortu da eta berarekin onuragarriak diren aldaketak planteatzera ekarri du. SGD eta berarengan oinarritutako ikasketa algoritmoak oso erabiliak dira elikadura zuzeneko sareak entrenatzeko eta planteatu diren liburutegi ikertzailea eta aldaketak baliagarriak izatea espero da, batez ere, ahalera mugatutako sistementzako erabiltzen diren sare neuronalak diseinatu eta entrenatzeko, ahalik eta elementu gutxien dituzten sare neuronalak sortzeko, baina eskatzen zaizkien baldintzak ondo beterik. Pozgarria da hain erabilia den algoritmo batentzako, dagoeneko modifikaketa ugari planteaturik dituena, beste ikuspuntu bat eman eta hau funtzionatzeko ahalmena edukitzea.

Falta izan den partean, irudien prozedurarako geruza konboluzionalak aztertu ahal izatea geratu egin da tesitik kanpo. Sare errekurtsiboak interesgarriak izango lirateke ere, ikerkuntza puntu bezala. Era honetan datuen sorkuntzarako sareen ikaskuntza ikertzea ahalbideratuko litzateke eta horietan ere hobekuntza posibilitateak ikusi. Ere bidean sare errekurtsiboak geratu egin dira. Bi sare mota hauek hurrengo pauso bezala geratzen dira, ea zer pausu gehio eman daitezkeen beraien ikasketaren ikerkuntzan.

3. Lanaren hedatzea

Egindako ikerkuntzen emaitzak era onean burutu dira ere. Hurrengoko hauek dira tesiaren zehar argitaratu diren artikulua:

- Differential Evolution Optimal Parameters Tuning with Artificial Neural Network[59].**
- A Data Augmentation-Based Techniques for Deep Learning Applied to CFD Simulations[91].**
- Dynamical Analysis of a Navigation Algorithm[93].**
- A Study of Learning Issues in Feedforward Neural Networks[56].**
- Neural architecture search for the estimation of relative positioning of the autonomous mobile robot[255]**
- Stability Analysis for Autonomous Vehicle Navigation Trained over Deep Deterministic Policy Gradient[60].**
- Convolutional Neural Network Predictions for Unsteady Reynolds-Averaged Navier-Stokes-Based Numerical Simulations[92].**
- Modification of Learning Ratio and Drop-Out for Stochastic Gradient Descendant Algorithm[57].**

Era honetan, tesian garatu den lana era egoki baten zabaltzea espero da. Honi gehituta, Databeers Euskadi seminarioan[329] aurkeztu egin zen egindako lana, Irudia 103-n ikusi daiteke egindako aurkezpena.



Irudia 103: Databeers seminarioan egindako aurkezpena.

4. Hurrengo pausuak

Eman daitezkeen hurrengo pausuen artean hurrengo lerroetan batzuk aipatuko dira:

- BasqueNet erabilia, adibide matematikoetatik aparte mundu fisikoko patriiak erabilia sareak entrenatu eta horien datuak erabilia, hemen planteatu diren esperimentuak egin eta ziurtatu. Nabigazio sistemak hasiera puntu bat izan daitezke, SLAM egiten duten sare neuronalak erabilia eta sentsoreetatik, edo beraien simulazio modeloetatik, datuak hartuta eta BasqueNet erabilia**
- BasqueNet zabaldu, geruza konbolutiboak gehiturik estrukturara, era honetan irudiekin lan egiteko aukera aurkezten da. Ere sare errekurtsiboak sortzeko aukera gehitzea aurrera pausu haundia izango lirateke, memoria behar duten aplikazioetara ere zuzenduko litzateke. Era honetan, erreminta osoago bat gertuko litzateke.**
- BasqueNet-en exekuzioa GPUra pasatzea, lanaren abiadura areagotuz. GPUak sare neuronalak lantzeko ingurumen aproposagoa da, CPUa baino, datuak paraleloan lantzeko aukerak hauen ikasketaren abiadura areagotuko luke, sare konplexuagoak landu ahal izateko eta ikasketan ematen diren beste efektu batzuk ere aztertzeko aukera emango luke.**
- Beste ikasketa algoritmoak aztertu eta egindako ikerketen bitartez hobekuntzak planteatu. Bai AdaGrad, RMSProp eta Adam motako algoritmoak aztertu, bai sorospen bidez lan egiten duten eta algoritmo genetikoak ere. Lehenengo kasuan, ikuskatutako ikasketa munduan oso erabiliak diren algoritmoak dira, lehen esan den moduan, eta oso interesgarria izango lirateke, BasqueNet-ekin ikertu ahal izatea, zer nolako ikasketa datuak agertzen diren ikusteko eta hobekuntzaren bat planteatzeko ere. Bigarren kasuan, BasqueNeten barruan aldaketa sakonak egitea ekarriko luke era horretako algoritmoak landu eta aztertu ahal izateko, baina benetan kasu guztietan lan egin dezakeen erreminta bat lortzea ekarriko luke.**

VII. Bibliografia

- [1] P. Dayal, «Semantic Segmentation Overview». Accedido: 7 de noviembre de 2020. [En línea]. Disponible en: <https://es.mathworks.com/videos/semantic-segmentation-overview-1510858047780.html>
- [2] L. Guo *et al.*, «Dual attention enhancement feature fusion network for segmentation and quantitative analysis of paediatric echocardiography», *Medical Image Analysis*, vol. 71, p. 102042, jul. 2021, doi: 10.1016/j.media.2021.102042.
- [3] J. Torres, *Python Deep Learning*, 1.0. 2020.
- [4] J. Torres, *Deep Learning Introducción práctica con Keras*, 3.0. en WATCH THIS SPACE, no. 1. Kindle Direct Publishing Amazon, 2018. [En línea]. Disponible en: www.JordiTorres.Barcelona/DeepLearning
- [5] Y. Li, H. Yang, J. Li, D. Chen, y M. Du, «EEG-based intention recognition with deep recurrent-convolution neural network: Performance and channel selection by Grad-CAM», *Neurocomputing*, vol. 415, pp. 225-233, nov. 2020, doi: 10.1016/j.neucom.2020.07.072.
- [6] K. Kläser *et al.*, «Imitation learning for improved 3D PET/MR attenuation correction», *Medical Image Analysis*, vol. 71, p. 102079, jul. 2021, doi: 10.1016/j.media.2021.102079.
- [7] K. O. Stanley y R. Miikkulainen, «Evolving Neural Networks through Augmenting Topologies», *Evolutionary Computation*, vol. 10, n.º 2, pp. 99-127, jun. 2002, doi: 10.1162/106365602320169811.
- [8] R. C. Martin, *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. en Martin, Robert C. Prentice Hall, 2018. [En línea]. Disponible en: <https://books.google.es/books?id=8ngAkAEACAAJ>
- [9] E. Gamma, R. Helm, R. E. Johnson, y J. Vlissides, *Design patterns: elements of reusable object-oriented software*. en Addison-Wesley professional computing series. Addison-Wesley, 2016. [En línea]. Disponible en: https://www.worldcat.org/title/design-patterns-elements-of-reusable-object-oriented-software/oclc/961356420&referer=brief_results
- [10] E. Evans, *Domain-Driven Design: Tackling Complexity in the Heart of Software*, 1.ª ed. Addison-Wesley, 2014. [En línea]. Disponible en: <http://www.worldcat.org/isbn/0321125215>
- [11] J. Bloch, *Effective Java: A Programming Language Guide*, 2nd Revised edition (REV). Addison-Wesley Longman, Amsterdam, 2008. [En línea]. Disponible en: http://www.amazon.de/Effective-Java-Programming-Language-Guide/dp/0321356683/ref=sr_1_1?ie=UTF8&qid=1303414280&sr=8-1
- [12] T. Rentsch, «Object oriented programming».
- [13] B. Meyer, *Object-Oriented Software Construction*, 2nd ed. Prentice Hall PTR, 2000. [En línea]. Disponible en: <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0136291554>
- [14] M. Fowler y K. Beck, *Refactoring: Improving the Design of Existing Code*, 1.ª ed. Addison-Wesley, 2013. [En línea]. Disponible en: <http://martinfowler.com/books/refactoring.html>
- [15] A. Watson, «Visual Modelling: past, present and future».
- [16] E. Planas y J. Cabot, «How are UML class diagrams built in practice? A usability study of two UML tools: Magicdraw and Papyrus», *Computer Standards & Interfaces*, vol. 67, p. 103363, ene. 2020, doi: 10.1016/j.csi.2019.103363.
- [17] R. P. De Lope, N. Medina-Medina, M. Urbieta, A. B. Lliteras, y A. Mora García, «A novel UML-based methodology for modeling adventure-based educational games», *Entertainment Computing*, vol. 38, p. 100429, may 2021, doi: 10.1016/j.entcom.2021.100429.
- [18] M. Abbas, R. Rioboo, C.-B. Ben-Yelles, y C. F. Snook, «Formal modeling and verification of UML Activity Diagrams (UAD) with FoCaLiZe», *Journal of Systems Architecture*, vol. 114, p. 101911, mar. 2021, doi: 10.1016/j.sysarc.2020.101911.
- [19] L. Addazi y F. Ciccozzi, «Blended graphical and textual modelling for UML profiles: A proof-of-concept implementation and experiment», *Journal of Systems and Software*, vol. 175, p. 110912, may 2021, doi: 10.1016/j.jss.2021.110912.
- [20] M. Mejhed Mkhinini, O. Labbani-Narsis, y C. Nicolle, «Combining UML and ontology: An exploratory survey», *Computer Science Review*, vol. 35, p. 100223, feb. 2020, doi: 10.1016/j.cosrev.2019.100223.
- [21] G. Bergström *et al.*, «Evaluating the layout quality of UML class diagrams using machine learning», *Journal of Systems and Software*, vol. 192, p. 111413, oct. 2022, doi: 10.1016/j.jss.2022.111413.

- [22] R. C. Martin, *Clean Agile*. en Robert C. Martin Series. Pearson Education, 2019. [En línea]. Disponible en: <https://books.google.es/books?id=QQ6vDwAAQBAJ>
- [23] R. C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*. en Robert C. Martin Series. Pearson Education, 2008. [En línea]. Disponible en: https://books.google.es/books?id=_i6bDeoCQzsC
- [24] R. Martin y an O. M. C. Safari, *Clean Craftsmanship: Disciplines, Standards, and Ethics*. Addison-Wesley Professional, 2021. [En línea]. Disponible en: <https://books.google.es/books?id=qC2izgEACAAJ>
- [25] R. C. Martin, *The Clean Coder: A Code of Conduct for Professional Programmers*. en Robert C. Martin series. Prentice Hall, 2011. [En línea]. Disponible en: <https://books.google.es/books?id=VQlvAQAAQBAJ>
- [26] «Software Garapen Arinaren aldeko Manifestua». Accedido: 29 de junio de 2023. [En línea]. Disponible en: <https://agilemanifesto.org/iso/eu/manifesto.html>
- [27] T. M. Mitchell, *Machine Learning*. en McGraw-Hill series in computer science. New York: McGraw-Hill, 1997.
- [28] V. Sze, Y.-H. Chen, T.-J. Yang, y J. S. Emer, «Efficient Processing of Deep Neural Networks: A Tutorial and Survey», *Proc. IEEE*, vol. 105, n.º 12, pp. 2295-2329, dic. 2017, doi: 10.1109/JPROC.2017.2761740.
- [29] A. L. Samuel, «Some Studies in Machine Learning Using the Game of Checkers», 1959.
- [30] M. Tavana, V. Hajipour, M. Alaghebandha, y D. Di Caprio, «A bi-objective hybrid vibration damping optimization model for synchronous flow shop scheduling problems», *Machine Learning with Applications*, vol. 11, p. 100445, mar. 2023, doi: 10.1016/j.mlwa.2022.100445.
- [31] A. Javeed, «A hybrid attention mechanism for multi-target entity relation extraction using graph neural networks», *Machine Learning with Applications*, vol. 11, p. 100444, mar. 2023, doi: 10.1016/j.mlwa.2022.100444.
- [32] M. Binkhonain y L. Zhao, «A machine learning approach for hierarchical classification of software requirements», *Machine Learning with Applications*, vol. 12, p. 100457, jun. 2023, doi: 10.1016/j.mlwa.2023.100457.
- [33] A. Schmidt, L. M. Ellsworth, J. H. Tilt, y M. Gough, «Application of deep convolutional networks for improved risk assessments of post-wildfire drinking water contamination», *Machine Learning with Applications*, vol. 11, p. 100454, mar. 2023, doi: 10.1016/j.mlwa.2023.100454.
- [34] Z. Chen, S. Czarnuch, E. Dove, y A. Astell, «Automated recognition of individual performers from de-identified video sequences», *Machine Learning with Applications*, vol. 11, p. 100450, mar. 2023, doi: 10.1016/j.mlwa.2023.100450.
- [35] K. D. Gupta, N. Sadman, A. Sadmanee, Md. K. Sarker, y R. George, «Behavioral recommendation engine driven by only non-identifiable user data», *Machine Learning with Applications*, vol. 11, p. 100442, mar. 2023, doi: 10.1016/j.mlwa.2022.100442.
- [36] W. Serrano, «Deep echo state networks in data marketplaces», *Machine Learning with Applications*, vol. 12, p. 100456, jun. 2023, doi: 10.1016/j.mlwa.2023.100456.
- [37] T. Krauth, A. Lafage, J. Morio, X. Olive, y M. Waltert, «Deep generative modelling of aircraft trajectories in terminal maneuvering areas», *Machine Learning with Applications*, vol. 11, p. 100446, mar. 2023, doi: 10.1016/j.mlwa.2022.100446.
- [38] S. Y. Appiah, E. K. Akowuah, V. C. Ikpo, y A. Dede, «Extremely randomised trees machine learning model for electricity theft detection», *Machine Learning with Applications*, vol. 12, p. 100458, jun. 2023, doi: 10.1016/j.mlwa.2023.100458.
- [39] J. Eiglsperger, F. Haselbeck, y D. G. Grimm, «ForeTiS: A comprehensive time series forecasting framework in Python», *Machine Learning with Applications*, vol. 12, p. 100467, jun. 2023, doi: 10.1016/j.mlwa.2023.100467.
- [40] E. Afzali y S. Muthukumarana, «Gradient-Free Kernel Conditional Stein Discrepancy goodness of fit testing», *Machine Learning with Applications*, vol. 12, p. 100463, jun. 2023, doi: 10.1016/j.mlwa.2023.100463.
- [41] E. S. Rahayu, E. M. Yuniarno, I. K. E. Purnama, y M. H. Purnomo, «Human activity classification using deep learning based on 3D motion feature», *Machine Learning with Applications*, vol. 12, p. 100461, jun. 2023, doi: 10.1016/j.mlwa.2023.100461.
- [42] B. Amirshahi y S. Lahmiri, «Hybrid deep learning and GARCH-family models for forecasting volatility of cryptocurrencies», *Machine Learning with Applications*, vol. 12, p. 100465, jun. 2023, doi: 10.1016/j.mlwa.2023.100465.

- [43] H. Kage, «Implementing associative memories by Echo State Network for the applications of natural language processing», *Machine Learning with Applications*, vol. 11, p. 100449, mar. 2023, doi: 10.1016/j.mlwa.2023.100449.
- [44] A. Sokolovsky, L. Arnaboldi, J. Bacardit, y T. Gross, «Interpretable trading pattern designed for machine learning applications», *Machine Learning with Applications*, vol. 11, p. 100448, mar. 2023, doi: 10.1016/j.mlwa.2023.100448.
- [45] V. Gopakumar, S. Pamela, y D. Samaddar, «Loss landscape engineering via Data Regulation on PINNs», *Machine Learning with Applications*, vol. 12, p. 100464, jun. 2023, doi: 10.1016/j.mlwa.2023.100464.
- [46] A. Rodriguez Dominguez, «Portfolio optimization based on neural networks sensitivities from assets dynamics respect common drivers», *Machine Learning with Applications*, vol. 11, p. 100447, mar. 2023, doi: 10.1016/j.mlwa.2022.100447.
- [47] D. Saha, T. M. Young, y J. Thacker, «Predicting firm performance and size using machine learning with a Bayesian perspective», *Machine Learning with Applications*, vol. 11, p. 100453, mar. 2023, doi: 10.1016/j.mlwa.2023.100453.
- [48] S. Hamdan, K. DuBray, J. Treutel, R. Paudyal, y K. Poudel, «Reducing MEG interference using machine learning», *Machine Learning with Applications*, vol. 12, p. 100462, jun. 2023, doi: 10.1016/j.mlwa.2023.100462.
- [49] R. Verachtert, O. Jeunen, y B. Goethals, «Scheduling on a budget: Avoiding stale recommendations with timely updates», *Machine Learning with Applications*, vol. 11, p. 100455, mar. 2023, doi: 10.1016/j.mlwa.2023.100455.
- [50] J. Saeedi y A. Giusti, «Semi-supervised visual anomaly detection based on convolutional autoencoder and transfer learning», *Machine Learning with Applications*, vol. 11, p. 100451, mar. 2023, doi: 10.1016/j.mlwa.2023.100451.
- [51] O. O. Awe, N. Dukhi, y R. Dias, «Shrinkage heteroscedastic discriminant algorithms for classifying multi-class high-dimensional data: Insights from a national health survey», *Machine Learning with Applications*, vol. 12, p. 100459, jun. 2023, doi: 10.1016/j.mlwa.2023.100459.
- [52] H. Ahmed, I. Traore, M. Mamun, y S. Saad, «Text augmentation using a graph-based approach and clonal selection algorithm», *Machine Learning with Applications*, vol. 11, p. 100452, mar. 2023, doi: 10.1016/j.mlwa.2023.100452.
- [53] C. Ringer *et al.*, «Time to Die 2: Improved in-game death prediction in Dota 2», *Machine Learning with Applications*, vol. 12, p. 100466, jun. 2023, doi: 10.1016/j.mlwa.2023.100466.
- [54] Z. Li, X. Wang, Z. Zhang, y V. Kindratenko, «ViCTer: A semi-supervised video character tracker», *Machine Learning with Applications*, vol. 12, p. 100460, jun. 2023, doi: 10.1016/j.mlwa.2023.100460.
- [55] C. Molnar, *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2019. [En línea]. Disponible en: <https://christophm.github.io/interpretable-ml-book/>
- [56] A. Teso-Fz-Betoño, E. Zulueta, M. Cabezas-Olivenza, D. Teso-Fz-Betoño, y U. Fernandez-Gamiz, «A Study of Learning Issues in Feedforward Neural Networks», *Mathematics*, vol. 10, n.º 17, 2022, doi: 10.3390/math10173206.
- [57] A. Teso-Fz-Betoño, E. Zulueta, M. Cabezas-Olivenza, U. Fernandez-Gamiz, y C. Botana-M-Ibarreta, «Modification of Learning Ratio and Drop-Out for Stochastic Gradient Descendant Algorithm», *Mathematics*, vol. 11, n.º 5, p. 1183, feb. 2023, doi: 10.3390/math11051183.
- [58] P. Kim, *MATLAB Deep Learning*. 2017.
- [59] M. Centeno-Telleria, E. Zulueta, U. Fernandez-Gamiz, D. Teso-Fz-Betoño, y A. Teso-Fz-Betoño, «Differential Evolution Optimal Parameters Tuning with Artificial Neural Network», *Mathematics*, vol. 9, n.º 4, p. 427, feb. 2021, doi: 10.3390/math9040427.
- [60] M. Cabezas-Olivenza, E. Zulueta, A. Sanchez-Chica, U. Fernandez-Gamiz, y A. Teso-Fz-Betoño, «Stability Analysis for Autonomous Vehicle Navigation Trained over Deep Deterministic Policy Gradient», *Mathematics*, vol. 11, n.º 1, p. 132, dic. 2022, doi: 10.3390/math11010132.
- [61] F. Giannakas, C. Troussas, I. Voyiatzis, y C. Sgouropoulou, «A deep learning classification framework for early prediction of team-based academic performance», *Applied Soft Computing*, vol. 106, p. 107355, jul. 2021, doi: 10.1016/j.asoc.2021.107355.
- [62] S. Shi *et al.*, «A Distributed Synchronous SGD Algorithm with Global Top-k Sparsification for Low Bandwidth Networks», en *2019 IEEE 39th International Conference on Distributed*

- Computing Systems (ICDCS)*, Dallas, TX, USA: IEEE, jul. 2019, pp. 2238-2247. doi: 10.1109/ICDCS.2019.00220.
- [63] G. Lai, F. Li, J. Feng, S. Cheng, y J. Cheng, «A LPSO-SGD algorithm for the Optimization of Convolutional Neural Network», en *2019 IEEE Congress on Evolutionary Computation (CEC)*, Wellington, New Zealand: IEEE, jun. 2019, pp. 1038-1043. doi: 10.1109/CEC.2019.8790362.
- [64] H. Yang, R. Zeng, G. Xu, y L. Zhang, «A network security situation assessment method based on adversarial deep learning», *Applied Soft Computing*, vol. 102, p. 107096, abr. 2021, doi: 10.1016/j.asoc.2021.107096.
- [65] F. Zou, L. Shen, Z. Jie, W. Zhang, y W. Liu, «A Sufficient Condition for Convergences of Adam and RMSProp», en *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, jun. 2019, pp. 11119-11127. doi: 10.1109/CVPR.2019.01138.
- [66] G. Litjens *et al.*, «A survey on deep learning in medical image analysis», *Medical Image Analysis*, vol. 42, pp. 60-88, dic. 2017, doi: 10.1016/j.media.2017.07.005.
- [67] L. J. B. Frey, «Adaptive dropout for training deep neural networks».
- [68] Y. Chen y Z. Yi, «Adaptive sparse dropout: Learning the certainty and uncertainty in deep neural networks», *Neurocomputing*, vol. 450, pp. 354-361, ago. 2021, doi: 10.1016/j.neucom.2021.04.047.
- [69] H. S. Suresha y S. S. Parthasarathy, «Alzheimer Disease Detection Based on Deep Neural Network with Rectified Adam Optimization Technique using MRI Analysis», en *2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAEECC)*, Bengaluru, India: IEEE, dic. 2020, pp. 1-6. doi: 10.1109/ICAEECC50550.2020.9339504.
- [70] G. Chen, C. K. Qu, y P. Gong, «Anomalous diffusion dynamics of learning in deep neural networks», *Neural Networks*, vol. 149, pp. 18-28, may 2022, doi: 10.1016/j.neunet.2022.01.019.
- [71] «Deep-neural-networks-regularization-for-structured-output-p_2018_Neurocomput.pdf».
- [72] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed. O'Reilly Media, Inc., 2019.
- [73] I. Goodfellow, Y. Bengio, y A. Courville, *Deep Learning*. MIT Press, 2016.
- [74] Z. Zhou, J. Y. Shin, S. R. Gurudu, M. B. Gotway, y J. Liang, «Active, continual fine tuning of convolutional neural networks for reducing annotation efforts», *Medical Image Analysis*, vol. 71, p. 101997, jul. 2021, doi: 10.1016/j.media.2021.101997.
- [75] A. R. Kalukin, «Automated Video Interpretability Assessment using Convolutional Neural Networks», en *2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, Washington, DC, USA: IEEE, oct. 2018, pp. 1-2. doi: 10.1109/AIPR.2018.8707423.
- [76] S. Brahimi, N. Ben Aoun, y C. Ben Amar, «Boosted Convolutional Neural Network for object recognition at large scale», *Neurocomputing*, vol. 330, pp. 337-354, feb. 2019, doi: 10.1016/j.neucom.2018.11.031.
- [77] T. Pradhan, P. Kumar, y S. Pal, «CLAVER: An integrated framework of convolutional layer, bidirectional LSTM with attention mechanism based scholarly venue recommendation», *Information Sciences*, vol. 559, pp. 212-235, jun. 2021, doi: 10.1016/j.ins.2020.12.024.
- [78] S. Mehta, C. Paunwala, y B. Vaidya, «CNN based Traffic Sign Classification using Adam Optimizer», en *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, Madurai, India: IEEE, may 2019, pp. 1293-1298. doi: 10.1109/ICCS45141.2019.9065537.
- [79] S. Y. Sen y N. Ozkurt, «Convolutional Neural Network Hyperparameter Tuning with Adam Optimizer for ECG Classification», en *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Istanbul, Turkey: IEEE, oct. 2020, pp. 1-6. doi: 10.1109/ASYU50717.2020.9259896.
- [80] B. Su, R. Li, y H. Zhang, «Evolving Deep Convolutional Neural Network for Intrusion Detection Based on NEAT», en *2020 23rd International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Okayama, Japan: IEEE, oct. 2020, pp. 1-6. doi: 10.1109/WPMC50192.2020.9309451.
- [81] R. V. Kumar Reddy, B. Srinivasa Rao, y K. P. Raju, «Handwritten Hindi Digits Recognition Using Convolutional Neural Network with RMSprop Optimization», en *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India: IEEE, jun. 2018, pp. 45-51. doi: 10.1109/ICCONS.2018.8662969.

- [82] L. Xu, J. Chen, y Y. Gan, «Head pose estimation with soft labels using regularized convolutional neural network», *Neurocomputing*, vol. 337, pp. 339-353, abr. 2019, doi: 10.1016/j.neucom.2018.12.074.
- [83] X. Yue, J. Wang, y W. Huang, «Hybrid fuzzy integrated convolutional neural network (HFICNN) for similarity feature recognition problem in abnormal netflow detection», *Neurocomputing*, vol. 415, pp. 332-346, nov. 2020, doi: 10.1016/j.neucom.2020.07.076.
- [84] C.-Y. Wang, T.-S. Ko, y C.-C. Hsu, «Interpreting Convolutional Neural Network for Real-time Volatile Organic Compounds Detection and Classification Using Optical Emission Spectroscopy of Plasma», *Analytica Chimica Acta*, p. 338822, 2021, doi: <https://doi.org/10.1016/j.aca.2021.338822>.
- [85] F. Letsch, D. Jirak, y S. Wermter, «Localizing salient body motion in multi-person scenes using convolutional neural networks», *Neurocomputing*, vol. 330, pp. 449-464, feb. 2019, doi: 10.1016/j.neucom.2018.11.048.
- [86] J. Yang y G. Yang, «Modified Convolutional Neural Network Based on Dropout and the Stochastic Gradient Descent Optimizer», *Algorithms*, vol. 11, n.º 3, p. 28, mar. 2018, doi: 10.3390/a11030028.
- [87] X. Liu, S. Bi, X. Ma, y J. Wang, «Multi-Instance Convolutional Neural Network for multi-shot person re-identification», *Neurocomputing*, vol. 337, pp. 303-314, abr. 2019, doi: 10.1016/j.neucom.2019.01.076.
- [88] M. Amirian y F. Schwenker, «Radial Basis Function Networks for Convolutional Neural Networks to Learn Similarity Distance Metric and Improve Interpretability», *IEEE Access*, vol. 8, pp. 123087-123097, 2020, doi: 10.1109/ACCESS.2020.3007337.
- [89] Z. Wei, C. Jung, y C. Su, «RegiNet: Gradient guided multispectral image registration using convolutional neural networks», *Neurocomputing*, vol. 415, pp. 193-200, nov. 2020, doi: 10.1016/j.neucom.2020.07.066.
- [90] A. Bahri, S. Ghofrani Majelan, S. Mohammadi, M. Noori, y K. Mohammadi, «Remote Sensing Image Classification via Improved Cross-Entropy Loss and Transfer Learning Strategy Based on Deep Convolutional Neural Networks», *IEEE Geosci. Remote Sensing Lett.*, vol. 17, n.º 6, pp. 1087-1091, jun. 2020, doi: 10.1109/LGRS.2019.2937872.
- [91] A. Abucide-Armas, K. Portal-Porras, U. Fernandez-Gamiz, E. Zulueta, y A. Teso-Fz-Betoño, «A Data Augmentation-Based Technique for Deep Learning Applied to CFD Simulations», *Mathematics*, vol. 9, n.º 16, p. 1843, ago. 2021, doi: 10.3390/math9161843.
- [92] A. Abucide-Armas, K. Portal-Porras, U. Fernandez-Gamiz, E. Zulueta, y A. Teso-Fz-Betoño, «Convolutional Neural Network Predictions for Unsteady Reynolds-Averaged Navier–Stokes-Based Numerical Simulations», *JMSE*, vol. 11, n.º 2, p. 239, ene. 2023, doi: 10.3390/jmse11020239.
- [93] M. Cabezas-Olivenza, E. Zulueta, A. Sánchez-Chica, A. Teso-Fz-Betoño, y U. Fernandez-Gamiz, «Dynamical Analysis of a Navigation Algorithm», *Mathematics*, vol. 9, n.º 23, p. 3139, dic. 2021, doi: 10.3390/math9233139.
- [94] D. Teso-Fz-Betoño, E. Zulueta, A. Sánchez-Chica, U. Fernandez-Gamiz, y A. Saenz-Aguirre, «Semantic Segmentation to Develop an Indoor Navigation System for an Autonomous Mobile Robot», *Mathematics*, vol. 8, p. 855, 2020, doi: 10.3390/math8050855.
- [95] A. Micheli, F. Portera, y A. Sperduti, «A preliminary empirical comparison of recursive neural networks and tree kernel methods on regression tasks for tree structured domains», *Neurocomputing*, vol. 64, pp. 73-92, mar. 2005, doi: 10.1016/j.neucom.2004.11.013.
- [96] D. Bani-Hani y M. Khasawneh, «A Recursive General Regression Neural Network (R-GRNN) Oracle for classification problems», *Expert Systems with Applications*, vol. 135, pp. 273-286, nov. 2019, doi: 10.1016/j.eswa.2019.06.018.
- [97] K. Khnissi, C. Ben Jabeur, y H. Seddik, «A smart mobile robot commands predictor using recursive neural network», *Robotics and Autonomous Systems*, vol. 131, p. 103593, sep. 2020, doi: 10.1016/j.robot.2020.103593.
- [98] J. Su, B. Zhang, D. Xiong, Y. Liu, y M. Zhang, «Alignment-consistent recursive neural networks for bilingual phrase embeddings», *Knowledge-Based Systems*, vol. 156, pp. 1-11, sep. 2018, doi: 10.1016/j.knosys.2018.05.003.
- [99] D. F. Wong, Y. Lu, y L. S. Chao, «Bilingual recursive neural network based data selection for statistical machine translation», *Knowledge-Based Systems*, vol. 108, pp. 15-24, sep. 2016, doi: 10.1016/j.knosys.2016.05.003.
- [100] J. Zhang *et al.*, «Daily runoff forecasting by deep recursive neural network», *Journal of Hydrology*, vol. 596, p. 126067, may 2021, doi: 10.1016/j.jhydrol.2021.126067.

- [101] L. Moreira y C. Guedes Soares, «Dynamic model of manoeuvrability using recursive neural networks», *Ocean Engineering*, vol. 30, n.º 13, pp. 1669-1697, sep. 2003, doi: 10.1016/S0029-8018(02)00147-6.
- [102] C. Bertinetto, C. Duce, A. Micheli, R. Solaro, A. Starita, y M. R. Tiné, «Evaluation of hierarchical structured representations for QSPR studies of small molecules and polymers by recursive neural networks», *Journal of Molecular Graphics and Modelling*, vol. 27, n.º 7, pp. 797-802, abr. 2009, doi: 10.1016/j.jmkgm.2008.12.001.
- [103] A. Parisi, F. Parisi, y D. Díaz, «Forecasting gold price changes: Rolling and recursive neural network models», *Journal of Multinational Financial Management*, vol. 18, n.º 5, pp. 477-487, dic. 2008, doi: 10.1016/j.mulfin.2007.12.002.
- [104] L. Wu y P. Baldi, «Learning to play Go using recursive neural networks», *Neural Networks*, vol. 21, n.º 9, pp. 1392-1400, nov. 2008, doi: 10.1016/j.neunet.2008.02.002.
- [105] J. Jiang, H. Wang, J. Xie, X. Guo, Y. Guan, y Q. Yu, «Medical knowledge embedding based on recursive neural network for multi-disease diagnosis», *Artificial Intelligence in Medicine*, vol. 103, p. 101772, mar. 2020, doi: 10.1016/j.artmed.2019.101772.
- [106] P. Baldi y M. Rosen-Zvi, «On the relationship between deterministic and probabilistic directed Graphical models: From Bayesian networks to recursive neural networks», *Neural Networks*, vol. 18, n.º 8, pp. 1080-1086, oct. 2005, doi: 10.1016/j.neunet.2005.07.007.
- [107] C. Bertinetto, C. Duce, A. Micheli, R. Solaro, A. Starita, y M. R. Tiné, «Prediction of the glass transition temperature of (meth)acrylic polymers containing phenyl groups by recursive neural network», *Polymer*, vol. 48, n.º 24, pp. 7121-7129, nov. 2007, doi: 10.1016/j.polymer.2007.09.043.
- [108] S. Avalos y J. M. Ortiz, «Recursive convolutional neural networks in a multiple-point statistics framework», *Computers & Geosciences*, vol. 141, p. 104522, ago. 2020, doi: 10.1016/j.cageo.2020.104522.
- [109] F. Biancofiore *et al.*, «Recursive neural network model for analysis and forecast of PM10 and PM2.5», *Atmospheric Pollution Research*, vol. 8, n.º 4, pp. 652-659, jul. 2017, doi: 10.1016/j.apr.2016.12.014.
- [110] M. Bianchini, M. Maggini, L. Sarti, y F. Scarselli, «Recursive neural networks learn to localize faces», *Pattern Recognition Letters*, vol. 26, n.º 12, pp. 1885-1895, sep. 2005, doi: 10.1016/j.patrec.2005.03.010.
- [111] D. Lin, G. Xu, W. Xu, Y. Wang, X. Sun, y K. Fu, «SCRSR: An efficient recursive convolutional neural network for fast and accurate image super-resolution», *Neurocomputing*, vol. 398, pp. 399-407, jul. 2020, doi: 10.1016/j.neucom.2019.02.067.
- [112] K. Zaporozhets, G. Bekoulis, J. Deleu, T. Demeester, y C. Develder, «Solving arithmetic word problems by scoring equations with recursive neural networks», *Expert Systems with Applications*, vol. 174, p. 114704, jul. 2021, doi: 10.1016/j.eswa.2021.114704.
- [113] M. Gori y A. Sperduti, «The loading problem for recursive neural networks», *Neural Networks*, vol. 18, n.º 8, pp. 1064-1079, oct. 2005, doi: 10.1016/j.neunet.2005.07.006.
- [114] H. Xu, H. He, Y. Zhang, L. Ma, y J. Li, «A comparative study of loss functions for road segmentation in remotely sensed road datasets», *International Journal of Applied Earth Observation and Geoinformation*, vol. 116, p. 103159, feb. 2023, doi: 10.1016/j.jag.2022.103159.
- [115] O. Gökkan y M. Kuntalp, «A new imbalance-aware loss function to be used in a deep neural network for colorectal polyp segmentation», *Computers in Biology and Medicine*, vol. 151, p. 106205, dic. 2022, doi: 10.1016/j.combiomed.2022.106205.
- [116] S. He, Q. Li, X. Li, y M. Zhang, «An optimized segmentation convolutional neural network with dynamic energy loss function for 3D reconstruction of lumbar spine MR images», *Computers in Biology and Medicine*, p. 106839, mar. 2023, doi: 10.1016/j.combiomed.2023.106839.
- [117] S. Fu, Y. Tian, J. Tang, y X. Liu, «Cost-sensitive learning with modified Stein loss function», *Neurocomputing*, vol. 525, pp. 57-75, mar. 2023, doi: 10.1016/j.neucom.2023.01.052.
- [118] G. Chen, Y. Hong, K. M. Huynh, y P.-T. Yap, «Deep learning prediction of diffusion MRI data with microstructure-sensitive loss functions», *Medical Image Analysis*, vol. 85, p. 102742, abr. 2023, doi: 10.1016/j.media.2023.102742.
- [119] L. Ge y L. Dou, «G-Loss: A loss function with gradient information for super-resolution», *Optik*, vol. 280, p. 170750, jun. 2023, doi: 10.1016/j.ijleo.2023.170750.
- [120] H. Shan, R. B. Vimieiro, L. R. Borges, M. A. C. Vieira, y G. Wang, «Impact of loss functions on the performance of a deep neural network designed to restore low-dose digital

- mammography», *Artificial Intelligence in Medicine*, p. 102555, abr. 2023, doi: 10.1016/j.artmed.2023.102555.
- [121] E. Ph. Kampitakis, P. Fafoutellis, G.-M. Oprea, y E. I. Vlahogianni, «Shared space multi-modal traffic modeling using LSTM networks with repulsion map and an intention-based multi-loss function», *Transportation Research Part C: Emerging Technologies*, vol. 150, p. 104104, may 2023, doi: 10.1016/j.trc.2023.104104.
- [122] Dr. Purwanto, C. Eswaran, y R. Logeswaran, «A Comparison of ARIMA, Neural Network and Linear Regression Models for the Prediction of Infant Mortality Rate», en *2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation*, Kota Kinabalu, Malaysia: IEEE, 2010, pp. 34-39. doi: 10.1109/AMS.2010.20.
- [123] A. Nicolson y K. K. Paliwal, «Deep learning for minimum mean-square error approaches to speech enhancement», *Speech Communication*, vol. 111, pp. 44-55, ago. 2019, doi: 10.1016/j.specom.2019.06.002.
- [124] Y. Sai, R. Jinxia, y L. Zhongxia, «Learning of Neural Networks Based on Weighted Mean Squares Error Function», en *2009 Second International Symposium on Computational Intelligence and Design*, Changsha, Hunan, China: IEEE, 2009, pp. 241-244. doi: 10.1109/ISCID.2009.67.
- [125] Z. Wang, T. Zhang, Y. Shao, y B. Ding, «LSTM-convolutional-BLSTM encoder-decoder network for minimum mean-square error approach to speech enhancement», *Applied Acoustics*, vol. 172, p. 107647, ene. 2021, doi: 10.1016/j.apacoust.2020.107647.
- [126] A. Kuh, «Mean squared error analysis of analog neural networks subject to drifting targets and noise», en *Conference Record of Thirty-Second Asilomar Conference on Signals, Systems and Computers (Cat. No.98CH36284)*, Pacific Grove, CA, USA: IEEE, 1998, pp. 683-684. doi: 10.1109/ACSSC.1998.750949.
- [127] W. N. Van Wieringen, «On the mean squared error of the ridge estimator of the covariance and precision matrix», *Statistics & Probability Letters*, vol. 123, pp. 88-92, abr. 2017, doi: 10.1016/j.spl.2016.12.002.
- [128] W. Ma, Y. Lei, X. Wang, y B. Chen, «Robust state of charge estimation of lithium-ion battery via mixture kernel mean p-power error loss LSTM with heap-based-optimizer», *Journal of Energy Chemistry*, vol. 80, pp. 768-784, may 2023, doi: 10.1016/j.jechem.2023.02.019.
- [129] Z. Diao, H. Jiang, y T. Shi, «A unified uncertainty network for tumor segmentation using uncertainty cross entropy loss and prototype similarity», *Knowledge-Based Systems*, vol. 246, p. 108739, jun. 2022, doi: 10.1016/j.knosys.2022.108739.
- [130] H. Chen *et al.*, «Adaptive Cross Entropy for ultrasmall object detection in Computed Tomography with noisy labels», *Computers in Biology and Medicine*, vol. 147, p. 105763, ago. 2022, doi: 10.1016/j.combiomed.2022.105763.
- [131] A. Motwani *et al.*, «Enhanced framework for COVID-19 prediction with computed tomography scan images using dense convolutional neural network and novel loss function», *Computers and Electrical Engineering*, vol. 105, p. 108479, ene. 2023, doi: 10.1016/j.compeleceng.2022.108479.
- [132] Y. Kim, Y. Lee, y M. Jeon, «Imbalanced image classification with complement cross entropy», *Pattern Recognition Letters*, vol. 151, pp. 33-40, nov. 2021, doi: 10.1016/j.patrec.2021.07.017.
- [133] A. Bahri, S. Ghofrani Majelan, S. Mohammadi, M. Noori, y K. Mohammadi, «Remote Sensing Image Classification via Improved Cross-Entropy Loss and Transfer Learning Strategy Based on Deep Convolutional Neural Networks», *IEEE Geosci. Remote Sensing Lett.*, vol. 17, n.º 6, pp. 1087-1091, jun. 2020, doi: 10.1109/LGRS.2019.2937872.
- [134] K. Hu *et al.*, «Retinal vessel segmentation of color fundus images using multiscale convolutional neural network with an improved cross-entropy loss function», *Neurocomputing*, vol. 309, pp. 179-191, oct. 2018, doi: 10.1016/j.neucom.2018.05.011.
- [135] Y. Xue, G. Jin, T. Shen, L. Tan, y L. Wang, «Template-guided frequency attention and adaptive cross-entropy loss for UAV visual tracking», *Chinese Journal of Aeronautics*, p. S1000936123001097, abr. 2023, doi: 10.1016/j.cja.2023.03.048.
- [136] Wang Tong y He Pi-Lian, «The classification algorithm based on the cross entropy rule and new activation function in fuzzy neural network», en *2005 International Conference on Machine Learning and Cybernetics*, Guangzhou, China: IEEE, 2005, pp. 4631-4635 Vol. 8. doi: 10.1109/ICMLC.2005.1527755.
- [137] M. Yeung, E. Sala, C.-B. Schönlieb, y L. Rundo, «Unified Focal loss: Generalising Dice and cross entropy-based losses to handle class imbalanced medical image segmentation»,

- Computerized Medical Imaging and Graphics*, vol. 95, p. 102026, ene. 2022, doi: 10.1016/j.compmedimag.2021.102026.
- [138] A. S. Bosman, A. Engelbrecht, y M. Helbig, «Visualising basins of attraction for the cross-entropy and the squared error neural network loss functions», *Neurocomputing*, vol. 400, pp. 113-136, ago. 2020, doi: 10.1016/j.neucom.2020.02.113.
- [139] N. Wang, «Fault line selection of power distribution system via improved bee colony algorithm based deep neural network», *Energy Reports*, vol. 8, pp. 43-53, nov. 2022, doi: 10.1016/j.egy.2022.10.070.
- [140] J. Sun, Q. Liu, Y. Wang, L. Wang, X. Song, y X. Zhao, «Five-Year Prognosis Model of Esophageal Cancer Based on Genetic Algorithm Improved Deep Neural Network», *IRBM*, vol. 44, n.º 3, p. 100748, jun. 2023, doi: 10.1016/j.irbm.2022.100748.
- [141] Z. Chiba, N. Abghour, K. Moussaid, A. El Omri, y M. Rida, «Intelligent approach to build a Deep Neural Network based IDS for cloud environment using combination of machine learning algorithms», *Computers & Security*, vol. 86, pp. 291-317, sep. 2019, doi: 10.1016/j.cose.2019.06.013.
- [142] K. O. Lye, S. Mishra, D. Ray, y P. Chandrashekar, «Iterative surrogate model optimization (ISMO): An active learning algorithm for PDE constrained optimization with deep neural networks», *Computer Methods in Applied Mechanics and Engineering*, vol. 374, p. 113575, feb. 2021, doi: 10.1016/j.cma.2020.113575.
- [143] G. Liao y L. Zhang, «Solving flows of dynamical systems by deep neural networks and a novel deep learning algorithm», *Mathematics and Computers in Simulation*, vol. 202, pp. 331-342, dic. 2022, doi: 10.1016/j.matcom.2022.06.004.
- [144] S. Nematzadeh, F. Kiani, M. Torkamaniafshar, y N. Aydin, «Tuning hyperparameters of machine learning algorithms and deep neural networks using metaheuristics: A bioinformatics study on biomedical and biological cases», *Computational Biology and Chemistry*, vol. 97, p. 107619, abr. 2022, doi: 10.1016/j.compbiolchem.2021.107619.
- [145] R. Sujatha, V. Mareeswari, J. M. Chatterjee, A. A. A. Mousa, y A. E. Hassanien, «A Bayesian Regularized Neural Network for Analyzing Bitcoin Trends», *IEEE Access*, vol. 9, pp. 37989-38000, 2021, doi: 10.1109/ACCESS.2021.3063243.
- [146] M. M. Almalki, E. S. Alaidarous, D. A. Maturi, M. A. Z. Raja, y M. Shoaib, «A Levenberg–Marquardt Backpropagation Neural Network for the Numerical Treatment of Squeezing Flow With Heat Transfer Model», *IEEE Access*, vol. 8, pp. 227340-227348, 2020, doi: 10.1109/ACCESS.2020.3044973.
- [147] K. Levenberg, «A method for the solution of certain non-linear problems in least squares», *Quart. Appl. Math.*, vol. 2, n.º 2, pp. 164-168, jul. 1944, doi: 10.1090/qam/10666.
- [148] K. Ucak y G. Oke, «Adaptive fuzzy PID controller based on online LSSVR», en *2012 International Symposium on Innovations in Intelligent Systems and Applications*, Trabzon, Turkey: IEEE, jul. 2012, pp. 1-7. doi: 10.1109/INISTA.2012.6247020.
- [149] A. Kr. Yadav, A. Azeem, A. Singh, H. Malik, y O. P. Rahi, «Application Research Based on Artificial Neural Network (ANN) to Predict No-Load Loss for Transformer's Design», en *2011 International Conference on Communication Systems and Network Technologies*, Katra, Jammu, India: IEEE, jun. 2011, pp. 180-183. doi: 10.1109/CSNT.2011.45.
- [150] A. Kr. Yadav, A. Singh, H. Malik, y A. Azeem, «Cost Analysis of Transformer's Main Material Weight with Artificial Neural Network (ANN)», en *2011 International Conference on Communication Systems and Network Technologies*, Katra, Jammu, India: IEEE, jun. 2011, pp. 184-187. doi: 10.1109/CSNT.2011.46.
- [151] Z. Chen, A. Huang, y X. Qiang, «Improved neural networks based on genetic algorithm for pulse recognition», *Computational Biology and Chemistry*, vol. 88, p. 107315, 2020, doi: <https://doi.org/10.1016/j.compbiolchem.2020.107315>.
- [152] N. T. Dieudonné, T. K. F. Armel, D. T. Hermann, A. K. C. Vidal, y T. René, «Optimization of Short-Term Forecast of Electric Power Demand in the city of Yaoundé-Cameroon by a hybrid model based on the combination of neural networks and econometric methods from a designed energy optimization algorithm», *Technological Forecasting and Social Change*, vol. 187, p. 122212, 2023, doi: <https://doi.org/10.1016/j.techfore.2022.122212>.
- [153] K. Ucak y G. Oke, «RBF neural network controller based on OLSSVR», en *2013 9th Asian Control Conference (ASCC)*, Istanbul, Turkey: IEEE, jun. 2013, pp. 1-6. doi: 10.1109/ASCC.2013.6606293.

- [154] L. Xiangmei y Q. Zhi, «The application of Hybrid Neural Network Algorithms in Intrusion Detection System», en *2011 International Conference on E-Business and E-Government (ICEE)*, Shanghai, China: IEEE, may 2011, pp. 1-4. doi: 10.1109/ICEBEG.2011.5882041.
- [155] D. Liu, W. He, y C. Zhang, «The research and optimization on levenberg-marquard algorithm in neural net», en *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chongqing, China: IEEE, mar. 2017, pp. 2442-2446. doi: 10.1109/IAEAC.2017.8054462.
- [156] D. Gutiérrez, A. Tapia Córdoba, y Á. Rodríguez del Nozal, *Algoritmos Genéticos con Python: Un enfoque práctico para resolver problemas de ingeniería*. 2020.
- [157] K. O. Stanley y R. Miikkulainen, «Efficient Reinforcement Learning through Evolving Neural Network Topologies», p. 9.
- [158] K. O. Stanley, R. Cornelius, R. Miikkulainen, y T. D'Silva, «Real-time Learning in the NERO Video Game», p. 2.
- [159] M. G. Cordeiro, P. B. S. Serafim, Y. L. B. Nogueira, C. A. Vidal, y J. B. Cavalcante Neto, «A Minimal Training Strategy to Play Flappy Bird Indefinitely with NEAT», en *2019 18th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, Rio de Janeiro, Brazil: IEEE, oct. 2019, pp. 21-28. doi: 10.1109/SBGames.2019.00014.
- [160] T. Kristo y N. U. Maulidevi, «Deduction of fighting game countermeasures using Neuroevolution of Augmenting Topologies», en *2016 International Conference on Data and Software Engineering (ICoDSE)*, Denpasar, Indonesia: IEEE, oct. 2016, pp. 1-6. doi: 10.1109/ICODSE.2016.7936127.
- [161] D. G. Nidorf, L. Barone, y T. French, «A comparative study of NEAT and XCS in Robocode», en *IEEE Congress on Evolutionary Computation*, Barcelona, Spain: IEEE, jul. 2010, pp. 1-8. doi: 10.1109/CEC.2010.5586087.
- [162] E. Papavasileiou y B. Jansen, «A comparison between FS-NEAT and FD-NEAT and an investigation of different initial topologies for a classification task with irrelevant features», en *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, Athens, Greece: IEEE, dic. 2016, pp. 1-8. doi: 10.1109/SSCI.2016.7850127.
- [163] A. Marzullo, C. Stamile, G. Terracina, F. Calimeri, y S. Van Huffel, «A tensor-based mutation operator for Neuroevolution of Augmenting Topologies (NEAT)», en *2017 IEEE Congress on Evolutionary Computation (CEC)*, Donostia, San Sebastián, Spain: IEEE, jun. 2017, pp. 681-687. doi: 10.1109/CEC.2017.7969376.
- [164] C. S. Ozveren, A. T. Sapeluk, y A. Birch, «An investigation into using neuro-evolution of Augmenting Topologies (NEAT) for short term load forecasting (STFL)», en *2014 49th International Universities Power Engineering Conference (UPEC)*, Cluj-Napoca, Romania: IEEE, sep. 2014, pp. 1-5. doi: 10.1109/UPEC.2014.6934819.
- [165] P. Caamano, F. Bellas, y R. J. Duro, «Augmenting the NEAT algorithm to improve its temporal processing capabilities», en *2014 International Joint Conference on Neural Networks (IJCNN)*, Beijing, China: IEEE, jul. 2014, pp. 1467-1473. doi: 10.1109/IJCNN.2014.6889488.
- [166] L. Bilgaeva, E. Sadykova, y V. Filippov, «Automatic Generation of Neural Network Topology to Forecast the International Integration Processes», en *2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, Vladivostok, Russia: IEEE, oct. 2019, pp. 1-7. doi: 10.1109/FarEastCon.2019.8934135.
- [167] J. Lowell, S. Grabkovsky, y K. Birger, «Comparison of NEAT and HyperNEAT Performance on a Strategic Decision-Making Problem», en *2011 Fifth International Conference on Genetic and Evolutionary Computing*, Kitakyushu, Japan: IEEE, ago. 2011, pp. 102-105. doi: 10.1109/ICGEC.2011.33.
- [168] B. I. Grisci, B. C. Feltes, y M. Dorn, «Microarray Classification and Gene Selection with FS-NEAT», en *2018 IEEE Congress on Evolutionary Computation (CEC)*, Rio de Janeiro: IEEE, jul. 2018, pp. 1-8. doi: 10.1109/CEC.2018.8477813.
- [169] J. S. Knapp y G. L. Peterson, «Natural Evolution Speciation for NEAT», en *2019 IEEE Congress on Evolutionary Computation (CEC)*, Wellington, New Zealand: IEEE, jun. 2019, pp. 1487-1493. doi: 10.1109/CEC.2019.8790153.
- [170] M. Gross, «Neat SIMD: Elegant vectorization in C++ by using specialized templates», en *2016 International Conference on High Performance Computing & Simulation (HPCS)*, Innsbruck, Austria: IEEE, jul. 2016, pp. 848-857. doi: 10.1109/HPCSim.2016.7568423.
- [171] K. L. Wallaschek, R. Klose, L. Almon, y M. Hollick, «NEAT-TCP: Generation of TCP Congestion Control through Neuroevolution of Augmenting Topologies», en *2020 IEEE*

- International Conference on Communications Workshops (ICC Workshops)*, Dublin, Ireland: IEEE, jun. 2020, pp. 1-6. doi: 10.1109/ICCWorkshops49005.2020.9145446.
- [172] R. Wen, Z. Guo, T. Zhao, X. Ma, Q. Wang, y Z. Wu, «Neuroevolution of augmenting topologies based muscuro-skeletal arm neurocontroller», en *2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Torino, Italy: IEEE, may 2017, pp. 1-6. doi: 10.1109/I2MTC.2017.7969727.
- [173] P. Reyes y M.-J. Escobar, «Neuroevolutive Algorithms for Learning Gaits in Legged Robots», *IEEE Access*, vol. 7, pp. 142406-142420, 2019, doi: 10.1109/ACCESS.2019.2944545.
- [174] D. Blasco, «Estudio del algoritmo NEAT aplicado al videojuego Pac-Man», Universitat Politècnica de Catalunya, 2019.
- [175] M. E. Taylor, S. Whiteson, y P. Stone, «Comparing evolutionary and temporal difference methods in a reinforcement learning domain», en *Proceedings of the 8th annual conference on Genetic and evolutionary computation - GECCO '06*, Seattle, Washington, USA: ACM Press, 2006, p. 1321. doi: 10.1145/1143997.1144202.
- [176] K. O. Stanley, B. D. Bryant, y R. Miikkulainen, «Evolving adaptive neural networks with and without adaptive synapses», en *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, Canberra, Australia: IEEE, 2003, pp. 2557-2564. doi: 10.1109/CEC.2003.1299410.
- [177] A. Perrusquía y W. Yu, «Identification and optimal control of nonlinear systems using recurrent neural networks and reinforcement learning: An overview», *Neurocomputing*, vol. 438, pp. 145-154, may 2021, doi: 10.1016/j.neucom.2021.01.096.
- [178] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, y J. Clune, «Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning», *arXiv:1712.06567 [cs]*, abr. 2018, Accedido: 14 de julio de 2021. [En línea]. Disponible en: <http://arxiv.org/abs/1712.06567>
- [179] O. Kramer, *Genetic Algorithm Essentials*, vol. 679. en *Studies in Computational Intelligence*, vol. 679. Cham: Springer International Publishing, 2017. doi: 10.1007/978-3-319-52156-5.
- [180] X. He, K. Zhao, y X. Chu, «AutoML: A survey of the state-of-the-art», *Knowledge-Based Systems*, vol. 212, p. 106622, ene. 2021, doi: 10.1016/j.knosys.2020.106622.
- [181] B. Xu, N. Wang, T. Chen, y M. Li, «Empirical Evaluation of Rectified Activations in Convolutional Network». 2015.
- [182] D.-A. Clevert, T. Unterthiner, y S. Hochreiter, «Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)». 2016.
- [183] G. Klambauer, T. Unterthiner, A. Mayr, y S. Hochreiter, «Self-Normalizing Neural Networks». arXiv, 7 de septiembre de 2017. Accedido: 15 de octubre de 2023. [En línea]. Disponible en: <http://arxiv.org/abs/1706.02515>
- [184] J. Duchi, E. Hazan, y Y. Singer, «Adaptive Subgradient Methods for Online Learning and Stochastic Optimization».
- [185] D. P. Kingma y J. Ba, «Adam: A Method for Stochastic Optimization». 2017.
- [186] A. H. Khan, X. Cao, S. Li, V. N. Katsikis, y L. Liao, «BAS-ADAM: an ADAM based approach to improve the performance of beetle antennae search optimizer», *IEEE/CAA J. Autom. Sinica*, vol. 7, n.º 2, pp. 461-471, mar. 2020, doi: 10.1109/JAS.2020.1003048.
- [187] N. Guan, L. Shan, C. Yang, W. Xu, y M. Zhang, «Delay Compensated Asynchronous Adam Algorithm for Deep Neural Networks», en *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*, Guangzhou: IEEE, dic. 2017, pp. 852-859. doi: 10.1109/ISPA/IUCC.2017.00130.
- [188] Z. Chang, Y. Zhang, y W. Chen, «Effective Adam-Optimized LSTM Neural Network for Electricity Price Forecasting», en *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China: IEEE, nov. 2018, pp. 245-248. doi: 10.1109/ICSESS.2018.8663710.
- [189] Z. Zhang, «Improved Adam Optimizer for Deep Neural Networks», en *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, Banff, AB, Canada: IEEE, jun. 2018, pp. 1-2. doi: 10.1109/IWQoS.2018.8624183.
- [190] S. Matsumura y T. Nakashima, «Incremental learning for SIRMs fuzzy systems by Adam method», en *2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems*

- (IFSA-SCIS), Otsu, Japan: IEEE, jun. 2017, pp. 1-4. doi: 10.1109/IFSA-SCIS.2017.8023307.
- [191] M. Naeem y S. A. Maged, «Linear Time Invariant State Space System Identification Using Adam Optimization», en *2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE)*, Aswan, Egypt: IEEE, feb. 2020, pp. 196-204. doi: 10.1109/ITCE48509.2020.9047808.
- [192] S. Jean, K. Cho, R. Memisevic, y Y. Bengio, «On Using Very Large Target Vocabulary for Neural Machine Translation». 2015.
- [193] M. Mohsin, H. Li, y H. B. Abdalla, «Optimization Driven Adam-Cuckoo Search-Based Deep Belief Network Classifier for Data Classification», *IEEE Access*, vol. 8, pp. 105542-105560, 2020, doi: 10.1109/ACCESS.2020.2999865.
- [194] K. Chen, H. Ding, y Q. Huo, «Parallelizing Adam Optimizer with Blockwise Model-Update Filtering», en *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain: IEEE, may 2020, pp. 3027-3031. doi: 10.1109/ICASSP40776.2020.9052983.
- [195] Y. Ma, R. Guo, M. Li, F. Yang, S. Xu, y A. Abubakar, «Supervised Descent Method for 2D Magnetotelluric Inversion using Adam Optimization», en *2019 International Applied Computational Electromagnetics Society Symposium - China (ACES)*, Nanjing, China: IEEE, ago. 2019, pp. 1-2. doi: 10.23919/ACES48530.2019.9060597.
- [196] J. Geng *et al.*, «A bidirectional dictionary LASSO regression method for online water quality detection in wastewater treatment plants», *Chemometrics and Intelligent Laboratory Systems*, vol. 237, p. 104817, jun. 2023, doi: 10.1016/j.chemolab.2023.104817.
- [197] D. M. Drukker y D. Liu, «A cluster plugin method for selecting the GLM lasso tuning parameters in models for unbalanced panel data», *Econometrics and Statistics*, p. S2452306222000132, mar. 2022, doi: 10.1016/j.ecosta.2022.02.006.
- [198] Y. Zhao y X. Huo, «Accelerate the warm-up stage in the Lasso computation via a homotopic approach», *Computational Statistics & Data Analysis*, vol. 184, p. 107747, ago. 2023, doi: 10.1016/j.csda.2023.107747.
- [199] Y. Zhang, M. Wu, Z. Zhang, H. Xu, Y. Zhou, y J. Liu, «An All-Arthroscopic Simple Double 360° Lasso Loop Technique for Suprapectoral Biceps Tenodesis», *Arthroscopy Techniques*, vol. 12, n.º 6, pp. e795-e800, jun. 2023, doi: 10.1016/j.eats.2023.02.008.
- [200] A. Mundakkal *et al.*, «Arthroscopic Subscapularis Tendon Repair Using the Lasso-Loop Technique Through Anterolateral Viewing Portal», *Arthroscopy Techniques*, vol. 12, n.º 1, pp. e83-e89, ene. 2023, doi: 10.1016/j.eats.2022.08.062.
- [201] M. G. Tsionas, «Combining data envelopment analysis and stochastic frontiers via a LASSO prior», *European Journal of Operational Research*, vol. 304, n.º 3, pp. 1158-1166, feb. 2023, doi: 10.1016/j.ejor.2022.04.029.
- [202] R. S. Koteeshwari y B. Malarkodi, «Compressive spectrum sensing for 5G cognitive radio networks – LASSO approach», *Heliyon*, vol. 8, n.º 6, p. e09621, jun. 2022, doi: 10.1016/j.heliyon.2022.e09621.
- [203] J. C. Álvarez Antón, P. J. García-Nieto, E. García-Gonzalo, M. González Vega, y C. Blanco Viejo, «Data-driven state-of-charge prediction of a storage cell using ABC/GBRT, ABC/MLP and LASSO machine learning techniques», *Journal of Computational and Applied Mathematics*, vol. 433, p. 115305, dic. 2023, doi: 10.1016/j.cam.2023.115305.
- [204] B.-J. Ma, H.-B. Ye, G.-Q. Meng, W. Zhao, Z. Ye, y J.-F. Ji, «Identification of key genes in spontaneous cerebral hemorrhage and prevention of disease damage: LASSO and SVM regression», *Preventive Medicine*, vol. 174, p. 107633, sep. 2023, doi: 10.1016/j.ympmed.2023.107633.
- [205] W. Shi, M. Machida, S. Yamada, T. Yoshida, Y. Hasegawa, y K. Okamoto, «Inverse estimation scheme of radioactive source distributions inside building rooms based on monitoring air dose rates using LASSO: Theory and demonstration», *Progress in Nuclear Energy*, vol. 162, p. 104792, ago. 2023, doi: 10.1016/j.pnucene.2023.104792.
- [206] R. Adamek, S. Smeekes, y I. Wilms, «Lasso inference for high-dimensional time series», *Journal of Econometrics*, vol. 235, n.º 2, pp. 1114-1143, ago. 2023, doi: 10.1016/j.jeconom.2022.08.008.
- [207] B. Uniejewski y K. Maciejowska, «LASSO principal component averaging: A fully automated approach for point forecast pooling», *International Journal of Forecasting*, p. S0169207022001352, oct. 2022, doi: 10.1016/j.ijforecast.2022.09.004.
- [208] W. Shi, M. Machida, S. Yamada, T. Yoshida, Y. Hasegawa, y K. Okamoto, «LASSO reconstruction scheme for radioactive source distributions inside reactor building rooms

- with spectral information and multi-radionuclide contaminated situations», *Annals of Nuclear Energy*, vol. 184, p. 109686, may 2023, doi: 10.1016/j.anucene.2023.109686.
- [209] V. Pontines y O. Rummel, «LIBOR meets machine learning: A Lasso regression approach to detecting data irregularities», *Finance Research Letters*, vol. 55, p. 103852, jul. 2023, doi: 10.1016/j.frl.2023.103852.
- [210] A. Massaro *et al.*, «Non-targeted authentication of black pepper using a local web platform: Development, validation and post-analytical challenges of a combined NIR spectroscopy and LASSO method», *Food Control*, vol. 145, p. 109477, mar. 2023, doi: 10.1016/j.foodcont.2022.109477.
- [211] A. Camehl, «Penalized estimation of panel vector autoregressive models: A panel LASSO approach», *International Journal of Forecasting*, vol. 39, n.º 3, pp. 1185-1204, jul. 2023, doi: 10.1016/j.ijforecast.2022.05.007.
- [212] S. Frndak *et al.*, «Reducing the complexity of high-dimensional environmental data: An analytical framework using LASSO with considerations of confounding for statistical inference», *International Journal of Hygiene and Environmental Health*, vol. 249, p. 114116, abr. 2023, doi: 10.1016/j.ijheh.2023.114116.
- [213] S. Song *et al.*, «Research on a working face gas concentration prediction model based on LASSO-RNN time series data», *Heliyon*, vol. 9, n.º 4, p. e14864, abr. 2023, doi: 10.1016/j.heliyon.2023.e14864.
- [214] F. Centofanti, M. Fontana, A. Lepore, y S. Vantini, «Smooth LASSO estimator for the Function-on-Function linear regression model», *Computational Statistics & Data Analysis*, vol. 176, p. 107556, dic. 2022, doi: 10.1016/j.csda.2022.107556.
- [215] M.-S. Yang y J. B. M. Benjamin, «Sparse possibilistic c-means clustering with Lasso», *Pattern Recognition*, vol. 138, p. 109348, jun. 2023, doi: 10.1016/j.patcog.2023.109348.
- [216] M. Czajkowski, K. Jurczuk, y M. Kretowski, «Steering the interpretability of decision trees using lasso regression - an evolutionary perspective», *Information Sciences*, vol. 638, p. 118944, ago. 2023, doi: 10.1016/j.ins.2023.118944.
- [217] P. Imbert, «The Lasso-Loop Technique: A Simple Way to Attach the Semitendinosus Tendon to an Endobutton Device», *Arthroscopy Techniques*, vol. 12, n.º 3, pp. e337-e341, mar. 2023, doi: 10.1016/j.eats.2022.11.012.
- [218] P. Kramlinger, U. Schneider, y T. Krivobokova, «Uniformly valid inference based on the Lasso in linear mixed models», *Journal of Multivariate Analysis*, p. 105230, ago. 2023, doi: 10.1016/j.jmva.2023.105230.
- [219] S. Jiao, Y. Zhang, Y. Xiao, X. Li, y M. Li, «Valuing carbon quota assets of power generation companies based on Lasso-Back propagation neural network», *Environmental Impact Assessment Review*, vol. 101, p. 107130, jul. 2023, doi: 10.1016/j.eiar.2023.107130.
- [220] J. Liang *et al.*, «VSOLassoBag: a variable-selection oriented LASSO bagging algorithm for biomarker discovery in omic-based translational research», *Journal of Genetics and Genomics*, vol. 50, n.º 3, pp. 151-162, mar. 2023, doi: 10.1016/j.jgg.2022.12.005.
- [221] S. Feld-Payet, V. Chiaruttini, J. Besson, y F. Feyel, «A new marching ridges algorithm for crack path tracking in regularized media», *International Journal of Solids and Structures*, vol. 71, pp. 57-69, oct. 2015, doi: 10.1016/j.ijsolstr.2015.04.043.
- [222] A. Fischer, S. Cellmer, y K. Nowel, «Assessment of the double-parameter iterative Tikhonov regularization for single-epoch measurement model-based precise GNSS positioning», *Measurement*, vol. 218, p. 113251, ago. 2023, doi: 10.1016/j.measurement.2023.113251.
- [223] P. Giordani y R. Rocci, «Candecomp/Parafac with ridge regularization», *Chemometrics and Intelligent Laboratory Systems*, vol. 129, pp. 3-9, nov. 2013, doi: 10.1016/j.chemolab.2013.08.002.
- [224] M. Mokhtia, M. Eftekhari, y F. Saberi-Movahed, «Dual-manifold regularized regression models for feature selection based on hesitant fuzzy correlation», *Knowledge-Based Systems*, vol. 229, p. 107308, oct. 2021, doi: 10.1016/j.knosys.2021.107308.
- [225] A. J. Schuster, A. P. D. R. Possebon, A. M. Bielemann, O. L. Chagas-Júnior, y F. Faot, «Effect of mandibular residual ridge regularization on peri-implant wound healing when narrow diameter implants are used as overdenture retainers», *The Journal of Prosthetic Dentistry*, vol. 128, n.º 4, pp. 648-655, oct. 2022, doi: 10.1016/j.prosdent.2020.12.029.
- [226] R. K. Tagayi, S. E. Ezahedi, J. Kim, y J. Kim, «Employment of relaxation times distribution with improved elastic net regularization for advanced impedance data analysis of a lithium-ion battery», *Journal of Energy Storage*, vol. 70, p. 107970, oct. 2023, doi: 10.1016/j.est.2023.107970.

- [227] S. Feld-Payet, V. Chiaruttini, J. Besson, y F. Feyel, «Erratum to: A new marching ridges algorithm for crack path tracking in regularized media [International Journal of Solids and Structures 71 (2015) 57–69]», *International Journal of Solids and Structures*, vol. 78-79, p. 219, ene. 2016, doi: 10.1016/j.ijsolstr.2015.10.005.
- [228] T. Dupré La Tour, M. Eickenberg, A. O. Nunez-Elizalde, y J. L. Gallant, «Feature-space selection with banded ridge regression», *NeuroImage*, vol. 264, p. 119728, dic. 2022, doi: 10.1016/j.neuroimage.2022.119728.
- [229] Y. Liu, S. Yu, y T. Lin, «Hessian regularization of deep neural networks: A novel approach based on stochastic estimators of Hessian trace», *Neurocomputing*, vol. 536, pp. 13-20, jun. 2023, doi: 10.1016/j.neucom.2023.03.017.
- [230] Z. Li, A. Pérez-Suay, G. Camps-Valls, y D. Sejdinovic, «Kernel dependence regularizers and Gaussian processes with applications to algorithmic fairness», *Pattern Recognition*, vol. 132, p. 108922, dic. 2022, doi: 10.1016/j.patcog.2022.108922.
- [231] Z. Chen, J. Hu, X. Qiu, y W. Jiang, «Kernel ridge regression-based TV regularization for motion correction of dynamic MRI», *Signal Processing*, vol. 197, p. 108559, ago. 2022, doi: 10.1016/j.sigpro.2022.108559.
- [232] K. A. Venkatesh, D. Mishra, y T. Manimozhi, «Model selection and regularization», en *Statistical Modeling in Machine Learning*, Elsevier, 2023, pp. 159-178. doi: 10.1016/B978-0-323-91776-6.24001-3.
- [233] S. Uysal, H. Uysal, y U. E. Ayten, «Novel regularization method for diffuse optical tomography inverse problem», *Optik*, vol. 261, p. 169095, jul. 2022, doi: 10.1016/j.ijleo.2022.169095.
- [234] M. Saccoccio, T. H. Wan, C. Chen, y F. Ciucci, «Optimal Regularization in Distribution of Relaxation Times applied to Electrochemical Impedance Spectroscopy: Ridge and Lasso Regression Methods - A Theoretical and Experimental Study», *Electrochimica Acta*, vol. 147, pp. 470-482, nov. 2014, doi: 10.1016/j.electacta.2014.09.058.
- [235] H. Li y H. Zhai, «Random Manifold Sampling and Joint Sparse Regularization for Multi-Label Feature Selection», *Big Data Research*, vol. 32, p. 100383, may 2023, doi: 10.1016/j.bdr.2023.100383.
- [236] H. Qing y J. Wang, «Regularized spectral clustering under the mixed membership stochastic block model», *Neurocomputing*, vol. 550, p. 126490, sep. 2023, doi: 10.1016/j.neucom.2023.126490.
- [237] D. Kraus y M. Stefanucci, «Ridge reconstruction of partially observed functional data is asymptotically optimal», *Statistics & Probability Letters*, vol. 165, p. 108813, oct. 2020, doi: 10.1016/j.spl.2020.108813.
- [238] H. Jiang, C. Tao, Y. Dong, y R. Xiong, «Robust low-rank multiple kernel learning with compound regularization», *European Journal of Operational Research*, vol. 295, n.º 2, pp. 634-647, dic. 2021, doi: 10.1016/j.ejor.2020.12.024.
- [239] Y. Cao, M. Jia, X. Zhao, X. Yan, y Z. Liu, «Semi-supervised machinery health assessment framework via temporal broad learning system embedding manifold regularization with unlabeled data», *Expert Systems with Applications*, vol. 222, p. 119824, jul. 2023, doi: 10.1016/j.eswa.2023.119824.
- [240] D. Karimova, R. Th. A. J. Leenders, M. Meijerink-Bosman, y J. Mulder, «Separating the wheat from the chaff: Bayesian regularization in dynamic social networks», *Social Networks*, vol. 74, pp. 139-155, jul. 2023, doi: 10.1016/j.socnet.2023.02.006.
- [241] H. Li, J. Dai, J. Xiao, X. Zou, T. Chen, y M. Holmose, «Spectral variable selection based on least absolute shrinkage and selection operator with ridge-adding homotopy», *Chemometrics and Intelligent Laboratory Systems*, vol. 221, p. 104487, feb. 2022, doi: 10.1016/j.chemolab.2021.104487.
- [242] H. Yildirim y M. Revan Özkale, «The performance of ELM based ridge regression via the regularization parameters», *Expert Systems with Applications*, vol. 134, pp. 225-233, nov. 2019, doi: 10.1016/j.eswa.2019.05.039.
- [243] Y. Chen, Y. Xiang, Z. Shi, J. Lu, y Y. Wang, «Tikhonov regularized penalty matrix construction method based on the magnitude of singular values and its application in near-field acoustic holography», *Mechanical Systems and Signal Processing*, vol. 170, p. 108870, may 2022, doi: 10.1016/j.ymsp.2022.108870.
- [244] S.-T. Huang, F. Xie, y J. Lederer, «Tuning-free ridge estimators for high-dimensional generalized linear models», *Computational Statistics & Data Analysis*, vol. 159, p. 107205, jul. 2021, doi: 10.1016/j.csda.2021.107205.

- [245] N. Tak y D. İnan, «Type-1 fuzzy forecasting functions with elastic net regularization», *Expert Systems with Applications*, vol. 199, p. 116916, ago. 2022, doi: 10.1016/j.eswa.2022.116916.
- [246] S. Ioffe y C. Szegedy, «Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift». 2015.
- [247] H. Taheri y Z. C. Xia, «SLAM; definition and evolution», *Engineering Applications of Artificial Intelligence*, vol. 97, p. 104032, 2021, doi: <https://doi.org/10.1016/j.engappai.2020.104032>.
- [248] M. Mustafa, A. Stancu, N. Delanoue, y E. Codres, «Guaranteed SLAM—An interval approach», *Robotics and Autonomous Systems*, vol. 100, pp. 160-170, 2018, doi: <https://doi.org/10.1016/j.robot.2017.11.009>.
- [249] G. Younes, D. Asmar, E. Shamma, y J. Zelek, «Keyframe-based monocular SLAM: design, survey, and future directions», *Robotics and Autonomous Systems*, vol. 98, pp. 67-88, 2017, doi: <https://doi.org/10.1016/j.robot.2017.09.010>.
- [250] S. Huang y G. Dissanayake, «Special Issue on Localization and Mapping in Challenging Environments», *Robotics and Autonomous Systems*, vol. 97, pp. 16-17, 2017, doi: <https://doi.org/10.1016/j.robot.2017.08.002>.
- [251] E. Yao, H. Zhang, H. Xu, H. Song, y G. Zhang, «Robust RGB-D visual odometry based on edges and points», *Robotics and Autonomous Systems*, vol. 107, pp. 209-220, 2018, doi: <https://doi.org/10.1016/j.robot.2018.06.009>.
- [252] D. Teso Fernández de Betoño, «Robotika kolaboratiboa nabigazio autonomoarekin bihurtuta prozesuak egiteko etengabeko lanean», Euskal Herriko Unibertsitatea, Vitoria-Gasteiz, 2020.
- [253] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, y S. Thrun, «Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing», en *2007 American Control Conference*, 2007, pp. 2296-2301. doi: 10.1109/ACC.2007.4282788.
- [254] M. Zhao *et al.*, «Faster Mean-shift: GPU-accelerated clustering for cosine embedding-based cell segmentation and tracking», *Medical Image Analysis*, vol. 71, p. 102048, jul. 2021, doi: 10.1016/j.media.2021.102048.
- [255] D. Teso-Fz-Betoño, E. Zulueta, A. Sanchez-Chica, U. Fernandez-Gamiz, A. Teso-Fz-Betoño, y J. M. Lopez-Guede, «Neural architecture search for the estimation of relative positioning of the autonomous mobile robot», *Logic Journal of the IGPL*, vol. 31, n.º 4, pp. 634-647, jul. 2023, doi: 10.1093/jigpal/jzac030.
- [256] X. Chen *et al.*, «Diverse data augmentation for learning image segmentation with cross-modality annotations», *Medical Image Analysis*, vol. 71, p. 102060, jul. 2021, doi: 10.1016/j.media.2021.102060.
- [257] Y. Zeng, Y. Gong, y J. Zhang, «Feature learning and patch matching for diverse image inpainting», *Pattern Recognition*, vol. 119, p. 108036, nov. 2021, doi: 10.1016/j.patcog.2021.108036.
- [258] J. Mellor, J. Turner, A. Storkey, y E. J. Crowley, «Neural Architecture Search without Training», *arXiv:2006.04647 [cs, stat]*, jun. 2020, Accedido: 9 de diciembre de 2020. [En línea]. Disponible en: <http://arxiv.org/abs/2006.04647>
- [259] H. Cai, L. Zhu, y S. Han, «ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware», *arXiv:1812.00332 [cs, stat]*, feb. 2019, Accedido: 21 de noviembre de 2020. [En línea]. Disponible en: <http://arxiv.org/abs/1812.00332>
- [260] H. Jin, Q. Song, y X. Hu, «Auto-Keras: An Efficient Neural Architecture Search System», en *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage AK USA: ACM, jul. 2019, pp. 1946-1956. doi: 10.1145/3292500.3330648.
- [261] M. Zhang *et al.*, «NAS-HRIS: Automatic Design and Architecture Search of Neural Network for Semantic Segmentation in Remote Sensing Images», *Sensors*, vol. 20, n.º 18, p. 5292, sep. 2020, doi: 10.3390/s20185292.
- [262] F. Doshi-Velez y B. Kim, *Towards A Rigorous Science of Interpretable Machine Learning*. 2017.
- [263] L. Lu, T. Pan, J. Zhao, y J. Yang, «Interpretability for Neural Networks from the Perspective of Probability Density», en *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, Xiamen, China: IEEE, dic. 2019, pp. 1502-1507. doi: 10.1109/SSCI44817.2019.9002817.

- [264] Z. Liu, C. Zhou, G. Hu, y C. Song, «Interpretability-Guided Convolutional Neural Networks for Seismic Fault Segmentation», en *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain: IEEE, may 2020, pp. 4312-4316. doi: 10.1109/ICASSP40776.2020.9053472.
- [265] J. P. Amorim, P. H. Abreu, M. Reyes, y J. Santos, «Interpretability vs. Complexity: The Friction in Deep Neural Networks», en *2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, United Kingdom: IEEE, jul. 2020, pp. 1-7. doi: 10.1109/IJCNN48605.2020.9206800.
- [266] M. Amirian y F. Schwenker, «Radial Basis Function Networks for Convolutional Neural Networks to Learn Similarity Distance Metric and Improve Interpretability», *IEEE Access*, vol. 8, pp. 123087-123097, 2020, doi: 10.1109/ACCESS.2020.3007337.
- [267] X. Yang, W. Wu, L. Jiao, C. Jiao, y Z. Jiao, «A deep fusion framework for unlabeled data-driven tumor recognition», *Pattern Recognition*, vol. 119, p. 108066, nov. 2021, doi: 10.1016/j.patcog.2021.108066.
- [268] S. Budd, E. C. Robinson, y B. Kainz, «A survey on active learning and human-in-the-loop deep learning for medical image analysis», *Medical Image Analysis*, vol. 71, p. 102062, jul. 2021, doi: 10.1016/j.media.2021.102062.
- [269] A. Li, F. Xiao, C. Zhang, y C. Fan, «Attention-based interpretable neural network for building cooling load prediction», *Applied Energy*, vol. 299, p. 117238, oct. 2021, doi: 10.1016/j.apenergy.2021.117238.
- [270] Z. Deng *et al.*, «Data-driven reconstruction of interpretable model for air supply system of proton exchange membrane fuel cell», *Applied Energy*, vol. 299, p. 117266, oct. 2021, doi: 10.1016/j.apenergy.2021.117266.
- [271] J. V. Sundgaard *et al.*, «Deep metric learning for otitis media classification», *Medical Image Analysis*, vol. 71, p. 102034, jul. 2021, doi: 10.1016/j.media.2021.102034.
- [272] X. Huang *et al.*, «A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability», *Computer Science Review*, vol. 37, p. 100270, ago. 2020, doi: 10.1016/j.cosrev.2020.100270.
- [273] S. Wang, Y. Yin, D. Wang, Z. Lv, Y. Wang, y Y. Jin, «An interpretable deep neural network for colorectal polyp diagnosis under colonoscopy», *Knowledge-Based Systems*, vol. 234, p. 107568, dic. 2021, doi: 10.1016/j.knosys.2021.107568.
- [274] B. Chen *et al.*, «An interpretable self-adaptive deep neural network for estimating daily spatially-continuous PM2.5 concentrations across China», *Science of The Total Environment*, vol. 768, p. 144724, may 2021, doi: 10.1016/j.scitotenv.2020.144724.
- [275] K. Yao, F. Cao, Y. Leung, y J. Liang, «Deep neural network compression through interpretability-based filter pruning», *Pattern Recognition*, vol. 119, p. 108056, nov. 2021, doi: 10.1016/j.patcog.2021.108056.
- [276] C. Jacobs, K. Glazebrook, A. K. Qin, y T. Collett, «Exploring the interpretability of deep neural networks used for gravitational lens finding with a sensitivity probe», *Astronomy and Computing*, vol. 38, p. 100535, ene. 2022, doi: 10.1016/j.ascom.2021.100535.
- [277] C. M. Dasari, S. Amilpur, y R. Bhukya, «Exploring variable-length features (motifs) for predicting binding sites through interpretable deep neural networks», *Engineering Applications of Artificial Intelligence*, vol. 106, p. 104485, nov. 2021, doi: 10.1016/j.engappai.2021.104485.
- [278] B. Zhang, Z. Dong, J. Zhang, y H. Lin, «Functional network: A novel framework for interpretability of deep neural networks», *Neurocomputing*, vol. 519, pp. 94-103, ene. 2023, doi: 10.1016/j.neucom.2022.11.035.
- [279] H. Xu y Z. Zhao, «NetBCE: An Interpretable Deep Neural Network for Accurate Prediction of Linear B-cell Epitopes», *Genomics, Proteomics & Bioinformatics*, vol. 20, n.º 5, pp. 1002-1012, oct. 2022, doi: 10.1016/j.gpb.2022.11.009.
- [280] Md. J. Alam y S. A. Fattah, «SR-AttNet: An Interpretable Stretch–Relax Attention based Deep Neural Network for Polyp Segmentation in Colonoscopy Images», *Computers in Biology and Medicine*, vol. 160, p. 106945, jun. 2023, doi: 10.1016/j.compbiomed.2023.106945.
- [281] Z. Salahuddin, H. C. Woodruff, A. Chatterjee, y P. Lambin, «Transparency of deep neural networks for medical image analysis: A review of interpretability methods», *Computers in Biology and Medicine*, vol. 140, p. 105111, ene. 2022, doi: 10.1016/j.compbiomed.2021.105111.
- [282] A. Perez-Suay *et al.*, «Interpretability of Recurrent Neural Networks in Remote Sensing», en *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing*

- Symposium*, Waikoloa, HI, USA: IEEE, sep. 2020, pp. 3991-3994. doi: 10.1109/IGARSS39084.2020.9323898.
- [283] X. Wen, W. Jiang, J. Zhan, X. Wang, y Z. He, «Interpretability Derived Backdoor Attacks Detection in Deep Neural Networks: Work-in-Progress», en *2020 International Conference on Embedded Software (EMSOFT)*, Shanghai, China: IEEE, sep. 2020, pp. 13-14. doi: 10.1109/EMSOFT51651.2020.9244019.
- [284] B. Ilan, A. Ranganath, J. Alvarez, S. Khatri, y R. Marcia, «Interpretability of ReLU for Inversion», en *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2022, pp. 1190-1195. doi: 10.1109/ICMLA55696.2022.00192.
- [285] Z. C. Lipton, «The Mythos of Model Interpretability», *CoRR*, vol. abs/1606.03490, 2016, [En línea]. Disponible en: <http://arxiv.org/abs/1606.03490>
- [286] N. L. Roux, M. Schmidt, y F. Bach, «A Stochastic Gradient Method with an Exponential Convergence Rate for Finite Training Sets». arXiv, 11 de marzo de 2013. Accedido: 14 de enero de 2023. [En línea]. Disponible en: <http://arxiv.org/abs/1202.6258>
- [287] A. Defazio, F. Bach, y S. Lacoste-Julien, «SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives». arXiv, 16 de diciembre de 2014. Accedido: 14 de enero de 2023. [En línea]. Disponible en: <http://arxiv.org/abs/1407.0202>
- [288] R. Johnson y T. Zhang, «Accelerating Stochastic Gradient Descent using Predictive Variance Reduction».
- [289] Z. Yang, «Adaptive stochastic conjugate gradient for machine learning», *Expert Systems with Applications*, vol. 206, p. 117719, nov. 2022, doi: 10.1016/j.eswa.2022.117719.
- [290] P. Wang y N. Zheng, «Convergence analysis of asynchronous stochastic recursive gradient algorithms», *Knowledge-Based Systems*, vol. 252, p. 109312, sep. 2022, doi: 10.1016/j.knosys.2022.109312.
- [291] Z. Li, T. Shang, X. Liu, P. Qian, y Y. Zhang, «Advanced multi-feedback stochastic parallel gradient descent wavefront correction in free-space optical communication», *Optics Communications*, p. 129268, ene. 2023, doi: 10.1016/j.optcom.2023.129268.
- [292] Q. Hu, L. Zhen, Y. Mao, S. Zhu, X. Zhou, y G. Zhou, «Adaptive stochastic parallel gradient descent approach for efficient fiber coupling», *Opt. Express*, vol. 28, n.º 9, p. 13141, abr. 2020, doi: 10.1364/OE.390762.
- [293] L. T. Phong y T. T. Phuong, «Differentially private stochastic gradient descent via compression and memorization», *Journal of Systems Architecture*, vol. 135, p. 102819, feb. 2023, doi: 10.1016/j.sysarc.2022.102819.
- [294] L. Roberts y E. Smyth, «A simplified convergence theory for Byzantine resilient stochastic gradient descent», *EURO Journal on Computational Optimization*, vol. 10, p. 100038, 2022, doi: 10.1016/j.ejco.2022.100038.
- [295] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, y J. Stainer, «Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent».
- [296] J.-K. Fang, C.-M. Fong, P. Yang, C.-K. Hung, W.-L. Lu, y C.-W. Chang, «AdaGrad Gradient Descent Method for AI Image Management», en *2020 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)*, Taoyuan, Taiwan: IEEE, sep. 2020, pp. 1-2. doi: 10.1109/ICCE-Taiwan49838.2020.9258085.
- [297] H. Wei, X. Zhang, y Z. Fang, «An Adaptive Quasi-Hyperbolic Momentum Method Based on AdaGrad+ Strategy», en *2022 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML)*, Xi'an, China: IEEE, oct. 2022, pp. 649-654. doi: 10.1109/ICICML57342.2022.10009763.
- [298] H. Li, X. Li, H. Dong, F. Han, y C. Wang, «Full-waveform inversion with adversarial losses via deep learning», *Journal of Applied Geophysics*, vol. 205, p. 104763, oct. 2022, doi: 10.1016/j.jappgeo.2022.104763.
- [299] K. Chakrabarti y N. Chopra, «Generalized AdaGrad (G-AdaGrad) and Adam: A State-Space Perspective», en *2021 60th IEEE Conference on Decision and Control (CDC)*, Austin, TX, USA: IEEE, dic. 2021, pp. 1496-1501. doi: 10.1109/CDC45484.2021.9682994.
- [300] C. Traoré y E. Pauwels, «Sequential convergence of AdaGrad algorithm for smooth convex optimization», *Operations Research Letters*, vol. 49, n.º 4, pp. 452-458, jul. 2021, doi: 10.1016/j.orl.2021.04.011.
- [301] M. A. Labbaf Khaniki, M. Behzad Hadi, y M. Manthouri, «Feedback Error Learning Controller based on RMSprop and Salp Swarm Algorithm for Automatic Voltage Regulator System», en *2020 10th International Conference on Computer and Knowledge Engineering*

- (ICCKE), Mashhad, Iran: IEEE, oct. 2020, pp. 425-430. doi: 10.1109/ICCKE50421.2020.9303718.
- [302] Y. Yu, L. Zhang, L. Chen, y Z. Qin, «Adversarial Samples Generation Based on RMSProp», en *2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP)*, Nanjing, China: IEEE, oct. 2021, pp. 1134-1138. doi: 10.1109/ICSIP52628.2021.9688946.
- [303] R. Zaheer y H. Shaziya, «A Study of the Optimization Algorithms in Deep Learning», en *2019 Third International Conference on Inventive Systems and Control (ICISC)*, Coimbatore, India: IEEE, ene. 2019, pp. 536-539. doi: 10.1109/ICISC44355.2019.9036442.
- [304] C. Fjellström y K. Nyström, «Deep learning, stochastic gradient descent and diffusion maps», *Journal of Computational Mathematics and Data Science*, vol. 4, p. 100054, ago. 2022, doi: 10.1016/j.jcmds.2022.100054.
- [305] T. Sun, K. Tang, y D. Li, «Gradient Descent Learning With Floats», *IEEE Trans. Cybern.*, vol. 52, n.º 3, pp. 1763-1771, mar. 2022, doi: 10.1109/TCYB.2020.2997399.
- [306] V. Ojha y G. Nicosia, «Backpropagation Neural Tree», *Neural Networks*, vol. 149, pp. 66-83, may 2022, doi: 10.1016/j.neunet.2022.02.003.
- [307] R. Senthil, B. Narayanan, y K. Velmurugan, «Develop the hybrid Adadelta Stochastic Gradient Classifier with optimized feature selection algorithm to predict the heart disease at earlier stage», *Measurement: Sensors*, vol. 25, p. 100602, feb. 2023, doi: 10.1016/j.measen.2022.100602.
- [308] H. Li *et al.*, «Adaptive Dropout Method Based on Biological Principles», *IEEE Trans. Neural Netw. Learning Syst.*, vol. 32, n.º 9, pp. 4267-4276, sep. 2021, doi: 10.1109/TNNLS.2021.3070895.
- [309] S. I. Mirzadeh, M. Farajtabar, y H. Ghasemzadeh, «Dropout as an Implicit Gating Mechanism For Continual Learning», en *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA: IEEE, jun. 2020, pp. 945-951. doi: 10.1109/CVPRW50498.2020.00124.
- [310] D. LeJeune, H. Javadi, y R. G. Baraniuk, «The Flip Side of the Reweighted Coin: Duality of Adaptive Dropout and Regularization». arXiv, 3 de enero de 2022. Accedido: 14 de febrero de 2023. [En línea]. Disponible en: <http://arxiv.org/abs/2106.07769>
- [311] Y. Ming, Y. Zhao, C. Wu, K. Li, y J. Yin, «Distributed and asynchronous Stochastic Gradient Descent with variance reduction», *Neurocomputing*, vol. 281, pp. 27-36, mar. 2018, doi: 10.1016/j.neucom.2017.11.044.
- [312] A. Sharma, «Guided parallelized stochastic gradient descent for delay compensation», *Applied Soft Computing*, vol. 102, p. 107084, abr. 2021, doi: 10.1016/J.ASOC.2021.107084.
- [313] J. Wang, H. Liang, y G. Joshi, «Overlap Local-SGD: An Algorithmic Approach to Hide Communication Delays in Distributed SGD», en *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain: IEEE, may 2020, pp. 8871-8875. doi: 10.1109/ICASSP40776.2020.9053834.
- [314] T. Kobayashi, «SCW-SGD: Stochastically Confidence-Weighted SGD», en *2020 IEEE International Conference on Image Processing (ICIP)*, Abu Dhabi, United Arab Emirates: IEEE, oct. 2020, pp. 1746-1750. doi: 10.1109/ICIP40778.2020.9190992.
- [315] X. Yang, X. Zheng, y H. Gao, «SGD-Based Adaptive NN Control Design for Uncertain Nonlinear Systems», *IEEE Trans. Neural Netw. Learning Syst.*, vol. 29, n.º 10, pp. 5071-5083, oct. 2018, doi: 10.1109/TNNLS.2018.2790479.
- [316] N. Zhang, W. Wu, y G. Zheng, «Convergence of Gradient Method With Momentum for Two-Layer Feedforward Neural Networks», *IEEE Trans. Neural Netw.*, vol. 17, n.º 2, pp. 522-525, mar. 2006, doi: 10.1109/TNN.2005.863460.
- [317] S. K. Lenka y A. G. Mohapatra, «Gradient Descent with Momentum based Neural Network Pattern Classification for the Prediction of Soil Moisture Content in Precision Agriculture», en *2015 IEEE International Symposium on Nanoelectronic and Information Systems*, Indore, India: IEEE, dic. 2015, pp. 63-66. doi: 10.1109/iNIS.2015.56.
- [318] N. Zhang, «Momentum algorithms in neural networks and the applications in numerical algebra», en *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*, Deng Feng, China: IEEE, ago. 2011, pp. 2192-2195. doi: 10.1109/AIMSEC.2011.6011205.
- [319] M. A. Wani y S. Afzal, «A New Framework for Fine Tuning of Deep Networks», en *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Cancun, Mexico: IEEE, dic. 2017, pp. 359-363. doi: 10.1109/ICMLA.2017.0-135.

- [320] Z. Zhang, P. Yang, X. Ren, Q. Su, y X. Sun, «Memorized sparse backpropagation», *Neurocomputing*, vol. 415, pp. 397-407, nov. 2020, doi: 10.1016/j.neucom.2020.08.055.
- [321] H. Tanaka, D. Kunin, D. L. K. Yamins, y S. Ganguli, «Pruning neural networks without any data by iteratively conserving synaptic flow». arXiv, 18 de noviembre de 2020. Accedido: 15 de julio de 2022. [En línea]. Disponible en: <http://arxiv.org/abs/2006.05467>
- [322] S. Surjanovic, D. Bingham, y S. Fraser, «Virtual Library Simulation Experiments: Test Function and Datasets». Accedido: 16 de diciembre de 2020. [En línea]. Disponible en: <https://www.sfu.ca/~ssurjano/optimization.html>
- [323] R. S. Jha, N. N. Jha, y M. M. Lele, «Stochastic gradient descent algorithm for the predictive modelling of grate combustion and boiler dynamics», *ISA Transactions*, p. S0019057822005651, nov. 2022, doi: 10.1016/j.isatra.2022.10.036.
- [324] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, y A. Anandkumar, «signSGD: Compressed Optimisation for Non-Convex Problems». arXiv, 7 de agosto de 2018. Accedido: 14 de enero de 2023. [En línea]. Disponible en: <http://arxiv.org/abs/1802.04434>
- [325] X. Xu y U. S. Kamilov, «SignProx: One-bit Proximal Algorithm for Nonconvex Stochastic Optimization», en *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, United Kingdom: IEEE, may 2019, pp. 7800-7804. doi: 10.1109/ICASSP.2019.8682059.
- [326] T. Sun y D. Li, «Sign Stochastic Gradient Descents without bounded gradient assumption for the finite sum minimization», *Neural Networks*, vol. 149, pp. 195-203, may 2022, doi: 10.1016/j.neunet.2022.02.012.
- [327] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, y R. R. Salakhutdinov, «Improving neural networks by preventing co-adaptation of feature detectors». arXiv, 3 de julio de 2012. Accedido: 14 de enero de 2023. [En línea]. Disponible en: <http://arxiv.org/abs/1207.0580>
- [328] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, y R. Salakhutdinov, «Dropout: A Simple Way to Prevent Neural Networks from Overfitting», *Journal of Machine Learning Research*, vol. 15, n.º 56, pp. 1929-1958, 2014.
- [329] «Databeers Euskadi | Aprende mientras compartes una cerveza.» Accedido: 30 de junio de 2023. [En línea]. Disponible en: <https://databeers.eus/>

Eranskinak

1. BaskeNet fitxategia

```
classdef BasqueNet
    properties
        Name
        NumberOfLayers
        Layers
        StopConditions
        LearningParameters
        AllWeightSum
        AzkenekoEpokakoModuloak
        AzkenekoIteraziokoModuloak
        Moduloak
        Entrenauta
        Analizador
        MediaValoresCapa
        rowsInput
        columnInput
        Equilibrar
        OraingoEpoka
        DropoutAdaptativo
    end

    methods
        function basqueNetObject = BasqueNet(name)
            %BASQUENET Constructor base, seteamos el nombre
            basqueNetObject.Name = name;
            basqueNetObject.NumberOfLayers = 0;
            basqueNetObject.Layers = Layer.empty;
            Entrenauta = false;
            basqueNetObject.MediaValoresCapa = struct();
            basqueNetObject.MediaValoresCapa(1).Capa= struct();
        end

        function basqueNetObject = addLayer(basqueNetObject,newLayer)
            %ADDLAYER Nos permite añadir una capa a la red
            basqueNetObject.NumberOfLayers = basqueNetObject.NumberOfLayers + 1;
            basqueNetObject.Layers(basqueNetObject.NumberOfLayers) = newLayer;
        end

        function basqueNetObject = removeLastLayer(basqueNetObject)
            %REMOVELASTLAYER Nos permite quitar la última capa introducida
            %delete(basqueNetObject.Layers(basqueNetObject.NumberOfLayers));
            basqueNetObject.NumberOfLayers = basqueNetObject.NumberOfLayers - 1;
            basqueNetObject.AzkenekoEpokakoModuloak = [];
        end

        function basqueNetObject = setLearningParameters(basqueNetObject, alpha,
            ikasketa, epocas, momentua, errorekinEpokak, miniBatch, entrenamendua, randomizeInput,
            normalizeLearning,dinamicAlpha, Equilibrar, DropoutAdaptativo)
            basqueNetObject.LearningParameters = Ikasketak(alpha, ikasketa, epocas,
            momentua, errorekinEpokak, miniBatch, entrenamendua, randomizeInput, normalizeLearning,
            dinamicAlpha);
            basqueNetObject.Equilibrar = Equilibrar;
            basqueNetObject.DropoutAdaptativo = DropoutAdaptativo;
        end
    end
end
```

```
end

function [output, basqueNetObject] = process(basqueNetObject, input)
    %PROCESS Función que tomando la entrada, la pasa por la red y
    %entrega la salida de la misma
    for iterator = 1: basqueNetObject.NumberOfLayers
        if(iterator == 1)
            [output, basqueNetObject.Layers(iterator).Name] = ...
                process(basqueNetObject.Layers(iterator).Name, input);
        else
            if (iterator >= 3)&& (isa(basqueNetObject.Layers(iterator-
2).Name, 'FullyConnected')) && (basqueNetObject.Layers(iterator-2).Name.HaveResidual ==
true)
                [output, basqueNetObject.Layers(iterator).Name] = ...
                    process(basqueNetObject.Layers(iterator).Name,
[output;basqueNetObject.Layers(iterator-2).Name.LastInput] );
            else
                [output, basqueNetObject.Layers(iterator).Name] = ...
                    process(basqueNetObject.Layers(iterator).Name, output);
            end
        end
    end
end

function [basqueNetObject, error, error_val] = ikasi(basqueNetObject, input,
target, ploteatu, validationIn, validationOut)

    %IKASI Función que realiza el aprendizaje de la red, procesando
    %las entradas y corrigiendo los pesos sinápticos en función del
    %error obtenido.
    if(nargin < 4)
        ploteatu = false;
    end
    if(nargin < 6)
        validate = false;
    else
        validate = true;
    end
    mean_error = [];
    mean_error_val = [];
    for superiter=1:basqueNetObject.LearningParameters.epocas
        for iter=1:basqueNetObject.NumberOfLayers
            basqueNetObject.MediavaloresCapa(superiter).Capa(iter).MaxCapa=0;
            basqueNetObject.MediavaloresCapa(superiter).Capa(iter).MinCapa=0;
            basqueNetObject.MediavaloresCapa(superiter).Capa(iter).MediaCapa=0;
            basqueNetObject.MediavaloresCapa(superiter).Capa(iter).Rango=0;
        end
    end
    basqueNetObject.Analizador = AnalizaCapas(basqueNetObject.NumberOfLayers,
basqueNetObject);
    %Realizamos esta operación para dejar estable la longitud de
    %out y error. Es como en C hacer
    %double out[epocas][longitudEntrada]
    [basqueNetObject.rowsInput, basqueNetObject.columnInput] = size(input);
    out =
zeros(basqueNetObject.LearningParameters.epocas,height(target),basqueNetObject.columnInp
ut);
```

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

```
error =
zeros(basqueNetObject.LearningParameters.epocas,height(target),basqueNetObject.columnInput);
    if(validate)
        error_val =
zeros(basqueNetObject.LearningParameters.epocas,length(validationIn));
    else
        error_val = zeros(basqueNetObject.LearningParameters.epocas,1);
    end
f = waitbar(0,'Ikasketa hazi egin da...', 'Name', 'Datuetatik ikasten');
cuentaError = 0;
figure('Name', 'Ikasketa datuak')
X = (1:basqueNetObject.NumberOfLayers);
Y = (1:basqueNetObject.columnInput);
for cuentaEpocas = 1:basqueNetObject.LearningParameters.epocas
    tic()
    basqueNetObject.OraingoEpoka = cuentaEpocas;
    if cuentaEpocas == basqueNetObject.LearningParameters.epocas
        basqueNetObject.Analizador =
incrementoEpoca(basqueNetObject.Analizador);
    end
    [basqueNetObject, out(cuentaEpocas,:), error(cuentaEpocas,:)] =...
        epokaBatIkasi(basqueNetObject, input, target);

    if(validate)
        error_val(cuentaEpocas,:) =
epokaBatBalidatu(basqueNetObject,validationIn, validationOut);
        if(abs(mean(error_val(cuentaEpocas,:))) >
abs(mean(error(cuentaEpocas,:))) &&
mean(error_val(cuentaEpocas,:))/mean(error(cuentaEpocas,:)) > 10)
            cuentaError = cuentaError +1;
        else
            cuentaError = 0;
        end
        if cuentaError == basqueNetObject.LearningParameters.erorekinEpokak
            break;
        end
    end
    if(plots)
        internalPlotear(basqueNetObject, out, error, target, cuentaEpocas);
    end

    tiempo = toc();

waitbar(cuentaEpocas/basqueNetObject.LearningParameters.epocas,f,sprintf('Epoka:%i / %i
Denbora Epokako: %12.6f s\nErrorea %12.9f BalidazioErrorea %12.9f',...

cuentaEpocas, ...

basqueNetObject.LearningParameters.epocas,...

tiempo,mean(error(cuentaEpocas,:), mean(error_val(cuentaEpocas,:)));
    mean_error(cuentaEpocas) = mean(error(cuentaEpocas,:));
    mean_error_val(cuentaEpocas) = mean(error_val(cuentaEpocas,:));
    capasEnPolares(basqueNetObject, X, Y);
    for iter=1:basqueNetObject.NumberOfLayers
        if(isa(basqueNetObject.Layers(iter).Name, 'FullyConnected'))
```

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

```
basqueNetObject.MediavaloresCapa(cuentaEpocas).Capa(iter).MaxCapa =
max(max(basqueNetObject.Layers(iter).Name.WeightsMatrix));

basqueNetObject.MediavaloresCapa(cuentaEpocas).Capa(iter).MinCapa =
min(min(basqueNetObject.Layers(iter).Name.WeightsMatrix));

basqueNetObject.MediavaloresCapa(cuentaEpocas).Capa(iter).MediaCapa =
mean(mean(basqueNetObject.Layers(iter).Name.WeightsMatrix));
        basqueNetObject.MediavaloresCapa(cuentaEpocas).Capa(iter).Rango
= range(basqueNetObject.Layers(iter).Name.WeightsMatrix, 'all');
        end
    end
end
```

```
internalPlotear(basqueNetObject, out, error, target, cuentaEpocas);
    Entrenauta = true;
    figure();
    loglog(mean_error)
    hold on
    loglog(mean_error_val)
    xlabel('Epoch');
    ylabel('Average Loss');
```

```
end
function JacobianoXNC = kalkulatuJakobianoBektorea(basqueNetObject,
kapaZenbakia)

    JacobianoXNC = kalkulatuJakobianoX(basqueNetObject, kapaZenbakia);

    JacobianoXNC = JacobianoXNC' *
basqueNetObject.Layers(kapaZenbakia).Name.LastInput';
end

function JacobianoXNCbias = kalkulatuJakobianoBektoreaBias(basqueNetObject,
kapaZenbakia)

    JacobianoXNC = kalkulatuJakobianoX(basqueNetObject, kapaZenbakia);

    JacobianoXNCbias = JacobianoXNC;
end
function basqueNetObject = insertarPenultimo(basqueNetObject, fc1, p1)
    lastLayer = basqueNetObject.Layers(basqueNetObject.NumberOfLayers);
    basqueNetObject.Layers(basqueNetObject.NumberOfLayers) = fc1;
    basqueNetObject.Layers(basqueNetObject.NumberOfLayers+1) = p1;
    basqueNetObject.Layers(basqueNetObject.NumberOfLayers+2) = lastLayer;
    basqueNetObject.NumberOfLayers = basqueNetObject.NumberOfLayers+2;
    basqueNetObject.AzkenekoEpokakoModuloak = [];
end
end
methods(Access = protected)
```

```
function [basqueNetObject, out, error] = epokaBatIkasi(basqueNetObject, input,
target)

    %EPOKABATIKASI Hace los pasos necesarios para aprender una
    %epoca completa.
    out = zeros(height(target),basqueNetObject.columnInput);
    error = zeros(height(target),basqueNetObject.columnInput);
    if(basqueNetObject.LearningParameters.randomizeInput)
        macrovector = [double(input);target];
```

```
    longitudMacroVector= size(macroVector);
    macroVector = macroVector(:,randperm(longitudMacroVector(2)));
    inputHeigh = height(input);
    targetHeigh = height(target);
    input=[];
    for iterador = 1:inputHeigh
        input=[input;macroVector(iterador,:)];
    end
    target=[];
    for iterador = 1:targetHeigh
        target=[target;macroVector(iterador+inputHeigh,:)];
    end
end
for iterator =1 : 1 : basqueNetObject.NumberOfLayers
    if(isa(basqueNetObject.Layers(iterator).Name, 'FullyConnected'))
        basqueNetObject.Layers(iterator).Name =
eraseLastModification(basqueNetObject.Layers(iterator).Name);
    end
end
for iterador = 1: basqueNetObject.columnInput
    basqueNetObject.Analizador =
incrementoPatron(basqueNetObject.Analizador, input(:,iterador));
    [out(:,iterador), basqueNetObject] = ...
        process(basqueNetObject,input(:,iterador));
    for iterator =1 : 1 : basqueNetObject.NumberOfLayers
        if(isa(basqueNetObject.Layers(iterator).Name, 'FullyConnected'))
            basqueNetObject.Analizador =
registrarJacobiano(basqueNetObject.Analizador,iterator,
basqueNetObject.Layers(iterator).Name.LastWeightMatrix,
basqueNetObject.Layers(iterator).Name.LastInput);
        else
            basqueNetObject.Analizador =
registrarJacobiano(basqueNetObject.Analizador,iterator,
basqueNetObject.Layers(iterator).Name.ProcessMatrix);
        end
    end
    error(:,iterador) = ...
        erreneaKalkulatu(basqueNetObject.LearningParameters.ikasketa,...
            out(:,iterador),...
            target(:,iterador));

    if(mod(iterador,basqueNetObject.LearningParameters.minibatch) == 0)
        basqueNetObject = ...
            azkenekoErroreazIkasi(basqueNetObject,...
                out(:,iterador),...
                target(:,iterador), true,
basqueNetObject.LearningParameters.minibatch);
    else if(iterador == basqueNetObject.columnInput)
        basqueNetObject = ...
            azkenekoErroreazIkasi(basqueNetObject,...
                out(:,iterador),...
                target(:,iterador), true,
mod(iterador,basqueNetObject.LearningParameters.minibatch));
    else
        basqueNetObject = ...
            azkenekoErroreazIkasi(basqueNetObject,...
                out(:,iterador),...
                target(:,iterador), false,
```

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

```
mod(iterator, basqueNetObject.LearningParameters.minibatch));
    end
    end
    basqueNetObject.AzkenekoEpokakoModuloak(iterator,:) =
basqueNetObject.AzkenekoIteraziokoModuloak;
    end
    for iterator = 1:basqueNetObject.NumberOfLayers
        if(isa(basqueNetObject.Layers(iterator).Name, 'FullyConnected'))
            basqueNetObject.AllWeightSum = basqueNetObject.AllWeightSum +...

basqueNetObject.Layers(iterator).Name.WeightSum;
        end
    end
    end
    function basqueNetObject = azkenekoErroreazIkasi(basqueNetObject, output, target,
aplicarMinibatch, elementosMiniBatch)
        %AZKENEKOERROREAZIKASI Función que realiza un único paso de
        %aprendizaje para una entrada concreta, con una salida
        %concreta.

        errorDeribatua = backward(basqueNetObject.LearningParameters.ikasketa,
output, target);
        basqueNetObject.Analizador =
registrarLoss(basqueNetObject.Analizador, errorDeribatua);
        partDev = errorDeribatua;
        for iterator = basqueNetObject.NumberOfLayers : -1:1
            if basqueNetObject.LearningParameters.normalizeLearning
                derivadaParaAprender = partDev / (norm(partDev) + realmin);
            else
                derivadaParaAprender = partDev;
            end
            if(isa(basqueNetObject.Layers(iterator).Name, 'FullyConnected'))
                basqueNetObject =
pisuakBerriztu(basqueNetObject, errorDeribatua, derivadaParaAprender, iterator,
aplicarMinibatch, elementosMiniBatch);
            end
            partDev = backward(basqueNetObject.Layers(iterator).Name, partDev);

            if (iterator >= 3) && isa(basqueNetObject.Layers(iterator-
2).Name, 'FullyConnected') && basqueNetObject.Layers(iterator-2).Name.HaveResidual ==
true
                for cuentaVector = basqueNetObject.Layers(iterator).Name.InputSize:
-1: basqueNetObject.Layers(iterator-1).Name.Size+1
                    partDev(:, cuentaVector) = [];
                end
            end
            basqueNetObject.AzkenekoIteraziokoModuloak(iterator) =
sacarModulo(basqueNetObject, partDev);
            end
            basqueNetObject.LearningParameters.mu =
basqueNetObject.LearningParameters.mu / 1.000001;
            end
            function error_val = epokaBatBalidatu(basqueNetObject, validationIn,
validationOut)
                out = zeros(1, length(validationIn));
                error_val = zeros(1, length(validationIn));
                for iterator = 1:length(validationIn)
                    [out(iterator), basqueNetObject] = ...
```

```
        process(basqueNetObject,validationIn(:,iterador));
        error_val(iterador) = ...
        erreareaKalkulatu(basqueNetObject.LearningParameters.ikasketa,...
            out(iterador),...
            validationOut(:,iterador));
    end
end
function internalPlotear(basqueNetObject, out,error, target, epoca)
    %INTERNALPLOTEAR Permite hacer un plot de la última epoca
    %procesada.
    figure('Name','Epoca: '+string(epoca));
    subplot(2,1,1);
    plot((1:length(target)),out(epoca, :),(1:length(target)),target)
    legend('Irteera','Objetiboa');
    title('Irteera')
    subplot(2,1,2);
    plot(error(epoca, :))
    title('Errorea')
    pause(3)
end
function basqueNetObject = pisuakBerriztu(basqueNetObject,errorDeribatua,
partDev , LayerPostua, aplicarMiniBatch, elementosMiniBatch)
    %PISUAKBERRIZTU Permite modificar los pesos sinápticos de una
    %capa fullyconnected.
    if (strcmp(basqueNetObject.LearningParameters.entrenamendua,'SGD'))
        [correccionPesos, correccionBias] = stochastic(basqueNetObject, partDev,
LayerPostua);
    else
        if (strcmp(basqueNetObject.LearningParameters.entrenamendua,'Levenberg-
Marquard'))
            [correccionPesos, correccionBias] =
LevenbergMarquard(basqueNetObject, LayerPostua, errorDeribatua);
        else
            [correccionPesos, correccionBias] = stochastic(basqueNetObject,
partDev);
        end
    end

    basqueNetObject.Analizador =
registrar(basqueNetObject.Analizador,LayerPostua, correccionBias, correccionPesos);

    correccionBias = dropoutBN(basqueNetObject, correccionBias, ...
        basqueNetObject.Layers(LayerPostua).Name.DropoutRate,
basqueNetObject.OraingoEpoka , LayerPostua);

    if basqueNetObject.Equilibrar
        correccionPesos = equilibrar(basqueNetObject, LayerPostua,
correccionPesos);
    else
        correccionPesos = dropoutBN(basqueNetObject, correccionPesos,...
            basqueNetObject.Layers(LayerPostua).Name.DropoutRate,
basqueNetObject.OraingoEpoka, LayerPostua);
    end

    if size(basqueNetObject.Layers(LayerPostua).Name.GuardadoMediaPesos) ~=
size(correccionPesos)
        basqueNetObject.Layers(LayerPostua).Name.GuardadoMediaPesos =
```

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

```
reshape(basqueNetObject.Layers(LayerPostua).Name.GuardadoMediaPesos,basqueNetObject.Layers(LayerPostua).Name.InputSize * basqueNetObject.Layers(LayerPostua).Name.OutputSize,1);
end

basqueNetObject.Layers(LayerPostua).Name.GuardadoMediaPesos =
basqueNetObject.Layers(LayerPostua).Name.GuardadoMediaPesos + correccionPesos;
basqueNetObject.Layers(LayerPostua).Name.GuardadoMediaBias =
basqueNetObject.Layers(LayerPostua).Name.GuardadoMediaBias + correccionBias;
if aplicarMiniBatch

    correccionPesos =
basqueNetObject.Layers(LayerPostua).Name.GuardadoMediaPesos ./elementosMiniBatch;
    correccionBias =
basqueNetObject.Layers(LayerPostua).Name.GuardadoMediaBias ./elementosMiniBatch;
    if(strcmp(basqueNetObject.LearningParameters.entrenamendua,'Levenberg-
Marquard'))
        correccionPesos = reshape(correccionPesos,
basqueNetObject.Layers(LayerPostua).Name.OutputSize,
basqueNetObject.Layers(LayerPostua).Name.InputSize);
        end
        momentoPesos = correccionPesos +
basqueNetObject.LearningParameters.momentua * ...
            basqueNetObject.Layers(LayerPostua).Name.LastMomentum;
        ikastekoPisuak = momentoPesos;

        basqueNetObject.Layers(LayerPostua).Name.LastModification =
basqueNetObject.Layers(LayerPostua).Name.LastModification +abs(ikastekoPisuak);
        basqueNetObject.Layers(LayerPostua).Name.WeightsMatrix = ...
            basqueNetObject.Layers(LayerPostua).Name.WeightsMatrix +
ikastekoPisuak;

        basqueNetObject.Layers(LayerPostua).Name.LastMomentum = momentoPesos;

        momentoBias = correccionBias +
basqueNetObject.LearningParameters.momentua * ...
            basqueNetObject.Layers(LayerPostua).Name.LastMomentumBias;
        ikastekoBias = momentoBias;

        basqueNetObject.Layers(LayerPostua).Name.BiasMatrix = ...
            basqueNetObject.Layers(LayerPostua).Name.BiasMatrix + ikastekoBias;
        basqueNetObject.Layers(LayerPostua).Name.LastModificationBias =
basqueNetObject.Layers(LayerPostua).Name.LastModificationBias + abs(ikastekoBias);
        basqueNetObject.Layers(LayerPostua).Name.LastMomentumBias = momentoBias;
        basqueNetObject.Layers(LayerPostua).Name.GuardadoMediaPesos =
zeros(basqueNetObject.Layers(LayerPostua).Name.OutputSize,basqueNetObject.Layers(LayerPostua).Name.InputSize);
        basqueNetObject.Layers(LayerPostua).Name.GuardadoMediaBias =
zeros(basqueNetObject.Layers(LayerPostua).Name.OutputSize,1);
    end

end

function ikastekoPisuak = dropoutBN(basqueNetObject, momentoPesos, dropoutRate,
epoka, LayerPostua)
%DROPOUTBN Crea la matriz de aprendizaje tras Dropout
if eq(basqueNetObject.DropoutAdaptativo, true)
    basqueNetObject.Layers(LayerPostua).Name.LastDropoutRate =
(LayerPostua)*(basqueNetObject.Layers(LayerPostua).Name.DropoutRate * exp(-0.1*(epoka-
```


Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

```
1));
    else
        basqueNetObject.Layers(LayerPostua).Name.LastDropuotRate =
basqueNetObject.Layers(LayerPostua).Name.DropoutRate;
    end
    [filas, columnas] = size(momentoPesos);
    matrizDrop = zeros(filas, columnas);
    numCandidatos = round(filas*columnas*abs(1-
basqueNetObject.Layers(LayerPostua).Name.LastDropuotRate));
    candidatosActivos = randperm(filas * columnas, numCandidatos);
    matrizDrop(candidatosActivos) = 1/(1-
basqueNetObject.Layers(LayerPostua).Name.LastDropuotRate);
    ikastekoPisuak = momentoPesos .* matrizDrop;

end
function moduloPesos = sacarModulo(basqueNetObject,partDev)
    moduloPesos = norm(partDev);
end

(basqueNetObject.NumberOfLayers / LayerPostua) * ...%% * (basqueNetObject.NumberOfLayers
/ LayerPostua)

function [correccionPesos, correccionBias] = stocastic(basqueNetObject,
errorDeribatua, LayerPostua)
    correccionPesos = basqueNetObject.LearningParameters.alpha * ...
        errorDeribatua' * ...
        basqueNetObject.Layers(LayerPostua).Name.LastInput';
    correccionBias = basqueNetObject.LearningParameters.alpha * errorDeribatua';
    if basqueNetObject.LearningParameters.dinamicAlpha
        correccionPesos = (basqueNetObject.NumberOfLayers / LayerPostua) *
correccionPesos;
        correccionBias = (basqueNetObject.NumberOfLayers / LayerPostua) *
correccionBias;
    end

end

function [correccionPesos, correccionBias] = levenbergMarquard(basqueNetObject,
LayerPostua, errorDeribatua)
    alpha = 10;
    J = kalkulatuJakobianoBektorea(basqueNetObject, LayerPostua);
    Jfila = reshape(J, 1,basqueNetObject.Layers(LayerPostua).Name.InputSize *
basqueNetObject.Layers(LayerPostua).Name.OutputSize);
    JBias = kalkulatuJakobianoBektoreaBias(basqueNetObject, LayerPostua);

    traspuesta = Jfila';
    correccionPesos = (traspuesta * Jfila + alpha * eye(length(Jfila))) \
(traspuesta *(errorDeribatua));
    traspuesta = JBias';
    correccionBias = (traspuesta * JBias + alpha * eye(length(JBias))) \
(traspuesta * errorDeribatua);
end

function capasEnPolares(basqueNetObject, X, Y)
    capas = contarFullyConnected(basqueNetObject);
    subplot(capas, 3,1)
    surf(X,Y, basqueNetObject.AzkenekoEpokakoModuloak);
    title('Deribatu partzialen moduluak');
    fullyLayer = 0;
```

```
for iterator = 1 : 1 : basqueNetObject.NumberOfLayers
    if(isa(basqueNetObject.Layers(iterator).Name, 'FullyConnected'))
        fullyLayer = fullyLayer + 1;
        subplot(capas, 3, fullyLayer * 2 + fullyLayer - 1);
        stri = strcat(string(iterator), '. Geruzako Pisuen modifikazio
akumul. ');
        if basqueNetObject.Layers(iterator).Name.InputSize > 1
            X = 1:1:basqueNetObject.Layers(iterator).Name.InputSize;
        else
            X = 1;
        end
        if basqueNetObject.Layers(iterator).Name.OutputSize > 1
            Y = 1: 1: basqueNetObject.Layers(iterator).Name.OutputSize;
        else
            Y = 1;
        end
        if basqueNetObject.Layers(iterator).Name.InputSize > 1 &&
basqueNetObject.Layers(iterator).Name.OutputSize > 1
            surf(X,
Y,basqueNetObject.Layers(iterator).Name.LastModification);
        else
            if basqueNetObject.Layers(iterator).Name.InputSize == 1
plot(basqueNetObject.Layers(iterator).Name.LastModification, 'b*');
            else
plot(basqueNetObject.Layers(iterator).Name.LastModification, 'b*');
            end
            end
            title(stri);
            stri = strcat(string(iterator), '. Geruzako Biasen modifikazio
akumul. ');
            subplot(capas, 3, fullyLayer * 2 + fullyLayer);
            plot(basqueNetObject.Layers(iterator).Name.LastModificationBias,
'b*');
            title(stri);
        end
    end
end
function capas = contarFullyConnected(basqueNetObject)
    capas = 0;
    for iterator = basqueNetObject.NumberOfLayers : -1:1
        if(isa(basqueNetObject.Layers(iterator).Name, 'FullyConnected'))
            capas = capas + 1;
        end
    end
end
function JacobianoX = kalkulatuJakobianoX(basqueNetObject, kapaZenbakia)
    if kapaZenbakia == basqueNetObject.NumberOfLayers
        JacobianoX = 1;
    end
    for iterador = basqueNetObject.NumberOfLayers: -1: kapaZenbakia + 1
        if iterador == basqueNetObject.NumberOfLayers
            JacobianoX = jakobianosLortu(basqueNetObject.Layers(iterador).Name);
        else
            if iterador ~= 1
                if isa(basqueNetObject.Layers(iterador-1).Name, 'FullyConnected')
```

```
        if basqueNetObject.Layers(iterador-1).Name.HaveResidual
            JacobianoX = JacobianoX(1:length(JacobianoX)-
basqueNetObject.Layers(iterador-1).Name.InputSize);
        end
    end
    end
    JacobianoX = JacobianoX *
jakobianosLortu(basqueNetObject.Layers(iterador).Name);
    end
    end
    function correccionPesos = equilibrar(basqueNetObject, LayerPostua,
correccionPesos)
        for dim =1: size(basqueNetObject.Layers(LayerPostua).Name.WeightsMatrix, 1)
            [B, I] = maxk(correccionPesos(dim,:), 5, 'ComparisonMethod','abs');
            for elementos = 1 :
size(basqueNetObject.Layers(LayerPostua).Name.WeightsMatrix, 2)
                if ismember(elementos, I)
                    correccionPesos(dim, elementos) = correccionPesos(dim,
elementos);
                else
                    correccionPesos(dim, elementos) = 0;
                end
            end
        end
    end
end
end
```

2. AnalizaCapas fitxategia

```
3. classdef AnalizaCapas
    %ANALIZACAPAS Summary of this class goes here
    % Detailed explanation goes here

    properties
        ModificadorPeso
        CurrentIndex
        CurrentEpoch
        regis
            NivelCorteLoss
            NivelCorteError
    end

    methods
        function analizarCapasObj = AnalizaCapas(numLayers,basquetObj)
            analizarCapasObj.ModificadorPeso = struct();
            analizarCapasObj.ModificadorPeso.mapa = struct();
            analizarCapasObj.ModificadorPeso.mapa.epoca= struct();
            analizarCapasObj.ModificadorPeso.mapa.epoca(1).patron = struct();
            analizarCapasObj.ModificadorPeso.mapa.epoca(1).patron(1).valor = 0;
            analizarCapasObj.ModificadorPeso.mapa.epoca(1).patron(1).Capa = struct();
            analizarCapasObj.ModificadorPeso.mapa.epoca(1).patron(1).Capa(1) = struct();
            for iter = 1:numLayers
```

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

```
        if isa(baskenetObj.Layers(iter).Name, 'FullyConnected')

anализarCapasObj.ModificadorPeso.mapa.epoca(1).patron(1).Capa(iter).modificacionPesos =
baskenetObj.Layers(iter).Name.WeightsMatrix;

anализarCapasObj.ModificadorPeso.mapa.epoca(1).patron(1).Capa(iter).modificacionBias =
baskenetObj.Layers(iter).Name.BiasMatrix;
        end
    end
    for iter = 1:numLayers
        if isa(baskenetObj.Layers(iter).Name, 'FullyConnected')

anализarCapasObj.ModificadorPeso.mapa.epoca(1).patron(1).Capa(iter).Jacobiano =
baskenetObj.Layers(iter).Name.LastWeightMatrix;

anализarCapasObj.ModificadorPeso.mapa.epoca(1).patron(1).Capa(iter).LastInput =
baskenetObj.Layers(iter).Name.LastInput;
            else

anализarCapasObj.ModificadorPeso.mapa.epoca(1).patron(1).Capa(iter).Jacobiano =
baskenetObj.Layers(iter).Name.ProcessMatrix;
                end

            end

        anализarCapasObj.ModificadorPeso.mapa.epoca(1).patron(1).Loss = 0;
        anализarCapasObj.CurrentIndex = 0;
        anализarCapasObj.CurrentEpoch = 0;
        anализarCapasObj.regis = false;
            anализarCapasObj.NivelCorteLoss = 0.5;
            anализarCapasObj.NivelCorteError = 0.5;
    end

    function anализarCapasObj = incrementoPatron(anализarCapasObj,patron)
        if anализarCapasObj.regis
            anализarCapasObj.CurrentIndex = anализarCapasObj.CurrentIndex + 1;

anализarCapasObj.ModificadorPeso.mapa.epoca(anализarCapasObj.CurrentEpoch).patron(analiz
arCapasObj.CurrentIndex).valor = patron;
                end
            end

    function anализarCapasObj = incrementoEpoca(anализarCapasObj)
        anализarCapasObj.regis = true;
        anализarCapasObj.CurrentEpoch = anализarCapasObj.CurrentEpoch + 1;
        anализarCapasObj.CurrentIndex = 0;
    end

    function anализarCapasObj = registrar(anализarCapasObj,LayerPostua,
correccionBias, correccionPesos)
        if anализarCapasObj.regis

anализarCapasObj.ModificadorPeso.mapa.epoca(anализarCapasObj.CurrentEpoch).patron(analiz
arCapasObj.CurrentIndex).Capa(LayerPostua).modificacionPesos = correccionPesos;

anализarCapasObj.ModificadorPeso.mapa.epoca(anализarCapasObj.CurrentEpoch).patron(analiz
arCapasObj.CurrentIndex).Capa(LayerPostua).modificacionBias = correccionBias;
                end
    end
```

Sare Neuronalen ikasketa analisia eta proposatutako hobekuntzak

```
end
function analizarCapasObj = registrarJacobiano(analizarCapasObj, LayerPostua,
Jacobiano, LastInput)
    if analizarCapasObj.regis

analizarCapasObj.ModificadorPeso.mapa.epoca(analizarCapasObj.CurrentEpoch).patron(analiz
arCapasObj.CurrentIndex).Capa(LayerPostua).Jacobiano = Jacobiano;
        if(nargin == 4)

analizarCapasObj.ModificadorPeso.mapa.epoca(analizarCapasObj.CurrentEpoch).patron(analiz
arCapasObj.CurrentIndex).Capa(LayerPostua).LastInput = LastInput;
            end
        end
    end

function analizarCapasObj = registrarLoss(analizarCapasObj, Loss)
    if analizarCapasObj.regis

analizarCapasObj.ModificadorPeso.mapa.epoca(analizarCapasObj.CurrentEpoch).patron(analiz
arCapasObj.CurrentIndex).Loss = Loss;
        end
    end

function analizarCapasObj = registrarError(analizarCapasObj, error)
    if analizarCapasObj.regis

analizarCapasObj.ModificadorPeso.mapa.epoca(analizarCapasObj.CurrentEpoch).patron(analiz
arCapasObj.CurrentIndex).error = error;
        end
    end

function [analizarCapasObj, sugerirParo, sugerirCambio] =
analizarEvolucionRed(analizarCapasObj)
    if analizarCapasObj.regis
        promedioLoss =
mean(abs(analizarCapasObj.ModificadorPeso.mapa.epoca(analizarCapasObj.CurrentEpoch).patr
on(1:length(patron)).Loss));
        promedioError =
mean(abs(analizarCapasObj.ModificadorPeso.mapa.epoca(analizarCapasObj.CurrentEpoch).patr
on(1:length(patron)).error));
        lossEsAlto = promedioLoss > NivelCorteLoss;
        errorEsAlto = promedioError > NivelCorteError;
        sugerirParo = ~lossEsAlto;
        sugerirCambio = sugerirParo && errorEsAlto;
    end
end
end
end
```

3. Layer fitzategia

```
classdef Layer
    %LAYER Summary of this class goes here
    % Detailed explanation goes here

    properties
        Name
    end
end
```

```
end

methods
function layerObject = Layer(name)
    %LAYER Construct an instance of this class
    % Detailed explanation goes here
    layerObject.Name = name;
end
function [output, layerObject] = process(layerObject, input)
    [output, layerObject] = internalProcess(layerObject, input);
end
function partDev = backward(layerObject, previousLayerBackward)
    partDev = internalBackward(layerObject, previousLayerBackward);
end
function kaparenDeribatua = jakobianosLortu(layerObject)
    kaparenDeribatua = internalJakobianoaLortu(layerObject);
end
end
methods(Access = protected)
function [output, layerObject] = internalProcess(layerObject, input)
    output = input;
    layerObject = layerObject;
end
function partDev = internalBackward(layerObject, previousLayerBackward)
    partDev = ones(1, length(previousLayerBackward));
end
function kaparenDeribatua = internalJakobianoaLortu(layerObject)
    kaparenDeribatua = ones(1, length(previousLayerBackward));
end
end
end
```

4. FullyConnected fitzategia

```
classdef FullyConnected < Layer
    %FULLYCONNECTED Summary of this class goes here
    % Detailed explanation goes here

    properties
        InputSize
        OutputSize
        weightsMatrix
        BiasMatrix
        Seed
        LastInput
        LastOutput
        LastMomentum
        LastMomentumBias
        weightSum
        DropOutRate
        LastDropoutRate
        GuardadoMediaPesos
        GuardadoMediaBias
        LastModification
        LastModificationBias
        LastweightMatrix
    end
end
```

```
HaveResidual
end

methods
function fullyConnectedObject = FullyConnected(name,inputSize, outputSize, seed,
dropOut, residual)
    if(nargin < 5)
        dropOut = 0;
    end
    if(nargin < 6)
        residual = false;
    end
    fullyConnectedObject@Layer(name);
    fullyConnectedObject.InputSize = inputSize;
    fullyConnectedObject.OutputSize = outputSize;
    fullyConnectedObject.Seed = seed;
    rng(fullyConnectedObject.Seed);
    fullyConnectedObject.WeightsMatrix = random('Uniform',-
1,1,outputSize,inputSize)./outputSize;
    fullyConnectedObject.BiasMatrix = random('Uniform',-
1,1,outputSize,1)./outputSize;
    fullyConnectedObject.LastMomentum = zeros(outputSize,inputSize);
    fullyConnectedObject.LastMomentumBias = zeros(outputSize,1);
    fullyConnectedObject.LastModification = zeros(outputSize,inputSize);
    fullyConnectedObject.LastModificationBias = zeros(outputSize,1);
    fullyConnectedObject.DropOutRate = dropOut;
    fullyConnectedObject.LastDropOutRate = dropOut;
    fullyConnectedObject.GuardadoMediaPesos = zeros(outputSize,inputSize);
    fullyConnectedObject.GuardadoMediaBias = zeros(outputSize,1);
    fullyConnectedObject.LastWeightMatrix = fullyConnectedObject.WeightsMatrix;
    fullyConnectedObject.HaveResidual = residual;
end
function fullyConnectedObject = eraseLastModification(fullyConnectedObject)
    fullyConnectedObject.LastModification =
zeros(fullyConnectedObject.OutputSize,fullyConnectedObject.InputSize);
    fullyConnectedObject.LastModificationBias =
zeros(fullyConnectedObject.OutputSize,1);
end
function fullyConnectedObject = resizeInputs(fullyConnectedObject, newSize)
    fullyConnectedObject.InputSize = newSize;
    fullyConnectedObject.WeightsMatrix = [];
    fullyConnectedObject.LastMomentum = [];
    fullyConnectedObject.LastModification = [];
    fullyConnectedObject.GuardadoMediaPesos = [];
    fullyConnectedObject.LastWeightMatrix = [];
    fullyConnectedObject.WeightsMatrix = random('Uniform',-
1,1,fullyConnectedObject.OutputSize,fullyConnectedObject.InputSize)./fullyConnectedObject.OutputSize;
    fullyConnectedObject.LastMomentum =
zeros(fullyConnectedObject.OutputSize,fullyConnectedObject.InputSize);
    fullyConnectedObject.LastModification =
zeros(fullyConnectedObject.OutputSize,fullyConnectedObject.InputSize);
    fullyConnectedObject.GuardadoMediaPesos =
zeros(fullyConnectedObject.OutputSize,fullyConnectedObject.InputSize);
    fullyConnectedObject.LastWeightMatrix = fullyConnectedObject.WeightsMatrix;
end
function fullyConnectedObject = resizeOutputs(fullyConnectedObject, newSize)
    fullyConnectedObject.OutputSize = newSize;
```

```
fullyConnectedObject.WeightsMatrix = [];  
fullyConnectedObject.BiasMatrix = [];  
fullyConnectedObject.LastMomentum = [];  
fullyConnectedObject.LastMomentumBias = [];  
fullyConnectedObject.LastModification = [];  
fullyConnectedObject.LastModificationBias = [];  
fullyConnectedObject.GuardadoMediaPesos = [];  
fullyConnectedObject.GuardadoMediaBias = [];  
fullyConnectedObject.LastWeightMatrix = [];  
  
fullyConnectedObject.WeightsMatrix = random('Uniform',-  
1,1,fullyConnectedObject.OutputSize,fullyConnectedObject.InputSize)./fullyConnectedObject.  
OutputSize;  
fullyConnectedObject.WeightsMatrix = fullyConnectedObject.WeightsMatrix .*  
eye(fullyConnectedObject.OutputSize,fullyConnectedObject.InputSize);  
fullyConnectedObject.BiasMatrix = random('Uniform',-  
1,1,fullyConnectedObject.OutputSize,1)./fullyConnectedObject.OutputSize;  
fullyConnectedObject.LastMomentum =  
zeros(fullyConnectedObject.OutputSize,fullyConnectedObject.InputSize);  
fullyConnectedObject.LastMomentumBias =  
zeros(fullyConnectedObject.OutputSize,1);  
fullyConnectedObject.LastModification =  
zeros(fullyConnectedObject.OutputSize,fullyConnectedObject.InputSize);  
fullyConnectedObject.LastModificationBias =  
zeros(fullyConnectedObject.OutputSize,1);  
fullyConnectedObject.GuardadoMediaPesos =  
zeros(fullyConnectedObject.OutputSize,fullyConnectedObject.InputSize);  
fullyConnectedObject.GuardadoMediaBias =  
zeros(fullyConnectedObject.OutputSize,1);  
fullyConnectedObject.LastWeightMatrix = fullyConnectedObject.WeightsMatrix;  
end  
end  
methods(Access = protected)  
function [output, layerObject] = internalProcess(layerObject, input)  
layerObject.LastWeightMatrix = layerObject.WeightsMatrix;  
if isequal(size(input),[layerObject.InputSize, 1])  
output = layerObject.WeightsMatrix * double(input) + ...  
layerObject.BiasMatrix;  
else  
output = zeros(layerObject.OutputSize, 1);  
end  
layerObject.LastInput =input;  
layerObject.LastOutput = output;  
end  
function partDev = internalBackward(layerObject, previouLayerBackward)  
partDev = previouLayerBackward * layerObject.LastWeightMatrix;  
end  
function kaparenDeribatua = internalJakobianoaLortu(layerObject)  
kaparenDeribatua = layerObject.LastWeightMatrix;  
end  
  
end  
end
```


5. ProcessingLayer fitxategia

```
classdef ProcessingLayer < Layer
    properties
        Size
        ProcessMatrix
    end
    methods
        function processingLayerObject = ProcessingLayer(name, size)
            processingLayerObject@Layer(name);
            processingLayerObject.Size = size;
            processingLayerObject.ProcessMatrix = zeros(size);
        end
        function processingLayerObject = resize(processingLayerObject,newSize)
            processingLayerObject.Size = newSize;
            processingLayerObject.ProcessMatrix = [];
            processingLayerObject.ProcessMatrix = zeros(newSize);
        end
    end
    methods(Access = protected)
        function partDev = internalBackward(layerObject, previouLayerBackward)
            partDev = previouLayerBackward * layerObject.ProcessMatrix;
        end
        function kaparenDeribatua = internalJakobianoaLortu(layerObject)
            kaparenDeribatua = layerObject.ProcessMatrix;
        end
    end
end
```

6. ReluLayer fitxategia

```
classdef ReluLayer < ProcessingLayer
    methods
        function reluLayerObject = ReluLayer(name, size)
            reluLayerObject@ProcessingLayer(name, size);
            reluLayerObject.Size = size;
        end
    end
    methods(Access = protected)
        function [output, layerObject] = internalProcess(layerObject, input)
            if(length(input) ~= layerObject.Size)
                output = zeros(layerObject.Size, 1);
            else
                output = zeros(layerObject.Size,1);
                for iterator=1:layerObject.Size
                    if input(iterator) < 0
                        output(iterator) = 0;
                        layerObject.ProcessMatrix(iterator, iterator) = 0;
                    else
                        output(iterator) = input(iterator);
                        layerObject.ProcessMatrix(iterator, iterator) = 1;
                    end
                end
            end
        end
    end
end
```

```
end  
end
```

7. SigmoidLayer fitxategia

```
classdef SigmoidLayer < ProcessingLayer  
    %DIGMOID Summary of this class goes here  
    % Detailed explanation goes here  
    methods  
        function objSigmoidLayer = SigmoidLayer(name,size)  
            %DIGMOID Construct an instance of this class  
            % Detailed explanation goes here  
            objSigmoidLayer@ProcessingLayer(name, size);  
        end  
    end  
    methods(Access = protected)  
        function [output, layerObject] = internalProcess(layerObject, input)  
            %PROCESS Apply the RELU operation on inputs  
            % Implementation of the class Layer method  
            output = zeros(layerObject.Size, 1);  
            for iterator=1:layerObject.Size  
                output(iterator) = 1/(1 + exp(-input(iterator)));  
                if layerObject.Size > 1  
                    layerObject.ProcessMatrix(iterator, iterator) = output(iterator) *  
(1 - output(iterator));  
                else  
                    layerObject.ProcessMatrix = output * (1 - output);  
                end  
            end  
        end  
    end  
end  
end
```

8. SoftMaxLayer fitxategia

```
classdef SoftMaxLayer < ProcessingLayer  
    methods  
        function softMaxLayerObject = SoftMaxLayer(name, size)  
            softMaxLayerObject@ProcessingLayer(name,size);  
        end  
    end  
    methods(Access = protected)  
        function [output, layerObject] = internalProcess(layerObject, input)  
            output = zeros(layerObject.Size, 1);  
            divisor = 0;  
            for iterator = 1: length(input)  
                divisor = divisor + exp(input(iterator));  
                output(iterator) = exp(input(iterator));  
            end  
            if divisor ~= 0  
                output = output / divisor;  
            end  
            layerObject.ProcessMatrix = diag(output) - (output .* output);  
        end  
    end
```

```
end  
end
```

9. Ikasketak fitzategia

```
classdef Ikasketak  
    properties  
        alpha  
        ikasketa  
        epocas  
        momentua  
        errorekinEpokak  
        minibatch  
        entrenamendua  
        mu  
        randomizeInput  
        normalizeLearning  
        dinamicAlpha;  
    end  
  
    methods  
        function ikasketaObjetua = Ikasketak(alpha, ikasketa, epocas, momentua,  
        errorekinEpokak, minibatch, entrenamendua, randomizeInput, normalizeLearning,  
        dinamicAlpha)  
            if(nargin < 1)  
                alpha = 0.01;  
            end  
            if(nargin < 2)  
                ikasketa = MSEIkasketa;  
            end  
            if(nargin < 3)  
                epocas = 10;  
            end  
            if(nargin < 4)  
                momentua = 0.02;  
            end  
            if(nargin < 5)  
                errorekinEpokak = 10;  
            end  
            if(nargin < 6)  
                miniBatch = 1;  
            end  
            if(nargin < 7)  
                entrenamendua = 'SGD';  
            end  
            if(nargin < 8)  
                randomizeInput = false;  
            end  
            if(nargin < 9)  
                normalizeLearning = false;  
            end  
            if(nargin < 10)  
                dinamicAlpha = false;  
            end  
            ikasketaObjetua.alpha = alpha;  
            ikasketaObjetua.ikasketa = ikasketa;  
            ikasketaObjetua.epocas = epocas;
```

```
        ikasketaObjetua.momentua = momentua;
        ikasketaObjetua.errorekinEpokak = errorekinEpokak;
        ikasketaObjetua.minibatch = miniBatch;
        ikasketaObjetua.entrenamendua = entrenamendua;
        ikasketaObjetua.mu = 1;
        ikasketaObjetua.randomizeInput = randomizeInput;
        ikasketaObjetua.normalizeLearning = normalizeLearning;
        ikasketaObjetua.dinamicAlpha = dinamicAlpha;
    end

end

end
```

10. IkasketaFuntzioa fitxategia

```
classdef IkasketaFuntzioa
    methods
        function ikasketaFuntzioObjetua = IkasketaFuntzioa()

            end

        function error = erroreakalkulatu(ikasfObj, irteera, esperotakoIrteera)
            error = barrukoErroreakalkulatu(ikasfObj, irteera, esperotakoIrteera);
        end

        function errorDeribatua = backward(ikasfObj, irteera, esperotakoIrteera)
            errorDeribatua = barrukoErrorDeribatua(ikasfObj, irteera,
            esperotakoIrteera);
        end
    end
    methods(Access = protected)
        function error = barrukoErroreakalkulatu(ikasfObj, irteera, esperotakoIrteera)
            error = esperotakoIrteera - irteera;
        end
        function errorDeribatua = barrukoErrorDeribatua(ikasfObj, irteera,
            esperotakoIrteera)
            errorDeribatua = ones(1,length(irteera));
        end
    end
end
end
```

11. MSEIkasketa fitxategia

```
classdef MSEIkasketa < IkasketaFuntzioa
    methods(Access = protected)
        function error = barrukoErroreakalkulatu(ikasfObj, irteera, esperotakoIrteera)
            error = (1/2) * (abs(esperotakoIrteera - irteera).^2);
        end
        function errorDeribatua = barrukoErrorDeribatua(ikasfObj, irteera,
            esperotakoIrteera)
            errorDeribatua = (esperotakoIrteera - irteera)';
        end
    end
end
end
```

12. EntropiaGurtzatutakoIkasketa fitzategia

```
classdef EntropiaGurtzatutakoIkasketa < IkasketaFuntzioa
    %ENTROPIAGURTZATUTAKOIKASKETA Summary of this class goes here
    % Detailed explanation goes here

    methods(Access = protected)
        function error = barrukoErroreaKalkulatu(ikasfObj,irteera, esperotakoIrteera)
            error = -sum( esperotakoIrteera .* log(irteera));
        end
        function errorDeribatua = barrukoErrorDeribatua(ikasfObj,irteera,
esperotakoIrteera)
            errorDeribatua = (-(esperotakoIrteera - irteera))';
        end
    end
end
end
```