ÎMBUNĂTĂȚIREA PERFORMANȚEI REȚELELOR NEURONALE PROFUNDE PRIN DEZVOLTAREA DE NOI FUNCȚII DE ACTIVARE

Teză de doctorat – Rezumat

pentru obținerea titlului științific de doctor la

Universitatea Politehnica Timișoara

în domeniul de doctorat Calculatoare și Tehnologia Informației

autor ing. Marina Adriana MERCIONI

conducător științific Prof.univ.dr.ing. Ștefan HOLBAN

În lucrarea intitulată „Îmbunătățirea performanței rețelelor neuronale profunde prin dezvoltarea de noi funcții de activare” sunt prezentate o serie variată de funcții de activare dezvoltate de-a lungul timpului în domeniul Învățării Profunde [1] care este un domeniu de actualitate foarte intens studiat.

Teza este structurată în cinci părți având în total 8 capitole și mai multe subcapitole (Fig.1. și Fig.2.) împărțite astfel:

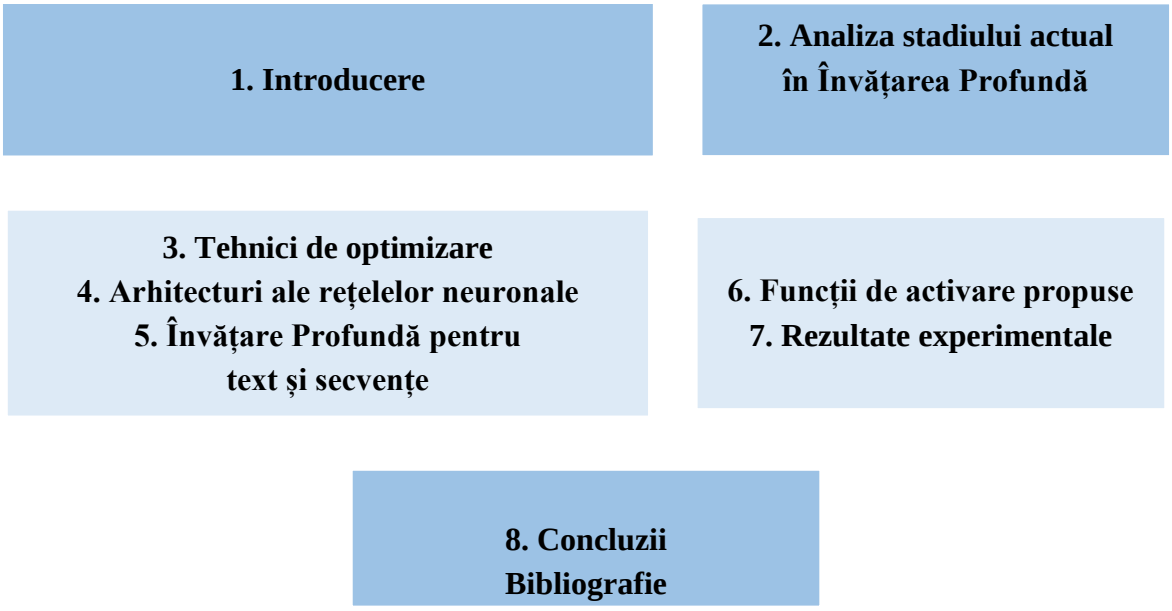

Fig.1. Arhitectura lucrării

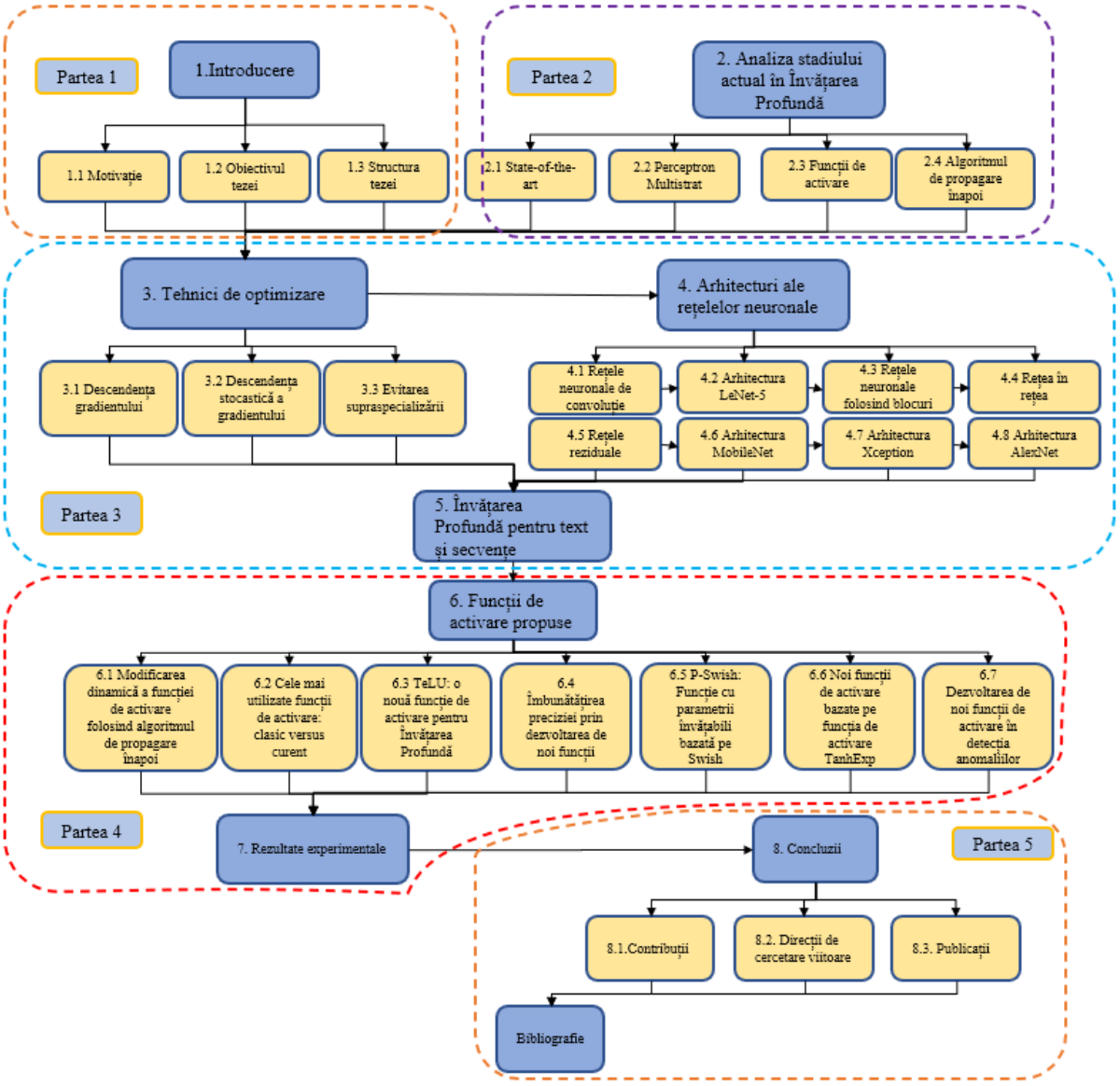

Fig.2. Arhitectura lucrării per subcapitole

Structura lucrării:

Prima parte conține capitolul 1 cu scopul de a prezenta motivația și obiectivul cercetării tratate în lucrare. Partea a doua conține analiza stadiului actual în domeniul funcțiilor de activare. Partea treia formată din capitolele 3, 4 și 5 în care sunt prezentate tehnicile de optimizare folosite în lucrare, diferite arhitecturi ale rețelelor neuronale pe baza cărora am făcut experimente și o scurtă prezentare a Învățării Profunde pentru text și secvențe. Partea a patra este partea principală a lucrării, aceasta conține capitolele 6 și 7 cu funcții de activare propuse și rezultatele experimentale. Ultima parte cuprinde capitolul 8 referitor la concluziile finale privind tratarea problemelor prezentate în lucrare precum și contribuțiile personale. Lucrarea se încheie cu direcții de cercetare viitoare, publicații și bibliografie.

Scopul lucrării constă în analiza detaliată și dezvoltarea unor noi funcții de activare în domeniul Învățării Profunde care să fie capabile să crească performanța atât pe sarcini din domeniul Vederii Artificiale [2] cât și pe sarcini din domeniul Procesării Limbajului Natural [3] dar și alte tipuri de sarcini precum detecția anomaliilor [4] sau predicții în seriile de timp.

Interesul pentru analiza Rețelelor Neuronale Artificiale prin folosirea unei funcții de

activare a crescut vertiginos în ultimii ani, devenind astfel un domeniu de cercetare foarte studiat, drept dovadă articolele existente în această direcție. Motiv pentru care în această teză doresc să subliniez importanța funcției de activare prin analiza mai multor seturi de date pentru diferite sarcini, pentru a vedea în ce mod impactează funcția de activare procesul de antrenare. În continuare voi prezenta, foarte succint, aspectele esențiale care constituie puncte cheie pe parcursul fiecărui capitol al tezei.

În ***primul capitol*** este prezentată motivația alegerii temei pentru teză, obiectivul urmărit și structura tezei. Motivația mea pornește de la faptul că fără funcții de activare, rețeaua neuronală poate să învețe doar sarcini de bază, astfel prin introducerea funcției de activare rețeaua este capabilă să învețe sarcini mai complexe. Astfel funcția de activare este un punct cheie în definirea arhitecturii rețelei neuronale și totodată funcția de activare este unul dintre parametri cei mai importanți pe care trebuie să îi alegem pentru a obține cu succes o performanță mai bună într-o rețea neuronală artificială. Plecând de rolul acesteia, pe care îl joacă în cadrul unei rețele neuronale, focusul meu s-a îndreptat spre studiul, analiza modului de funcționare, al avantajelor și dezavantajelor funcțiilor de activare pentru a găsi funcția care se mapează cel mai bine pe tipul de sarcină, aducând cele mai bune performanțe. O proprietate decisivă care determină performanța unei funcții de activare este dată de faptul că este sau nu netedă. [5] Proprietate analizată și de mine în cadrul acestei teze prin comparație cu alte funcții precum: funcția de activare tangentă hiperbolică (*tanh*) [6], funcția ReLU (*rectified linear unit*) [7], funcția de activare Swish [8], și așa mai departe. Această proprietate conferă funcției capacitatea să aibă derivate continue, până la un anumit ordin specificat. Acest lucru implică faptul că funcția este diferențiabilă continuu, altfel zis prima derivată există peste tot și este continuă. De asemenea, datorită faptului că acest domeniu este în continuă dezvoltare, o altă direcție a fost dezvoltarea de noi funcții de activare care să aducă îmbunătățiri la arhitectura rețelelor neuronale artificiale.

Această teză are drept obiectiv investigarea utilizării diferitelor funcții de activare care sunt parte integrantă a rețelelor neuronale artificiale în domeniul Învățării Profunde și dezvoltarea de noi funcții de activare care sa aducă îmbunătățiri în timpul antrenării rețelelor. Acest domeniu având mare succes și fiind foarte studiat în ultimii ani datorită disponibilității din punct de vedere hardware, dispunând de unități de procesare grafică (*Graphics processing unit – GPU*) precum și de volume tot mai mari de date (*Big Data*). Pentru atingerea acestui obiectiv, am definit următoarele sarcini:

- analiza stadiului actual în domeniu şi identificarea unor noi direcţii de cercetare în contextul domeniului funcțiilor de activare, care s-au dezvoltat de-a lungul timpului, fiind o arie de cercetare care se dezvoltă în mod continuu precum și modul de corelare al funcțiilor de activare cu cerințele sarcinii, tipul arhitecturii dar și tipul de date.
- identificarea unor funcții de activare care aduc îmbunătățiri substanțiale și conduc la o convergență rapidă a rețelelor neuronale artificiale. Această sarcină se bazează pe analiza funcțiilor de activare şi selectarea celei mai potrivite funcții pe baza datelor, care constituie de asemenea un potenţial de exploatare şi impact.
- implementarea mai multor tipuri de modele şi evaluarea acestora în funcţie de anumite metrici specifice precum precizie, acurateţe, funcție de cost, eroarea absolută medie, ș.a.m.d.

În ***al doilea capitol*** care corespunde și celei de-a ***doua părți***, îmi focusez atenția asupra prezentării state of the art și a situației actuale în domeniu, arhitectura de tip Perceptron

Multistrat, funcții de activare și algoritmul de Propagare Înapoi. Inteligența Artificială (*Artificial Intelligence* - AI) constituie punctul cheie în dezvoltarea tehnologică, permițând calculatoarelor să modeleze lumea reală la un grad foarte ridicat, obținând rezultate comparabile cu realitatea. S-au înregistrat progrese semnificative în domeniul rețelelor neuronale – un număr ridicat și suficient cât să ne atragă atenția în această direcție. Pentru a realiza acest lucru, avem nevoie de o cantitate mare de informații despre tot ce ne înconjoară, informații care trebuie să fie stocate în calculator. Practic se dă o anumită formă acestor date pe care calculatoarele o pot folosi pentru a răspunde la întrebări, oferind răspunsuri tot mai îmbunătățite pe măsură ce modelul dispune de un volum tot mai mare de informații. Pentru a generaliza acest context mulți cercetători s-au orientat spre algoritmi de învățare pentru a stoca o cantitate de informații într-un timp scurt. S-au făcut multe progrese pentru înțelegerea și îmbunătățirea acestor algoritmi de învățare, dar Inteligența Artificială constituie în continuare o provocare.

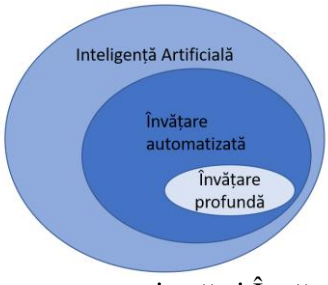

Fig.3. Inteligență artificială, Învățare automatizată și Învățare profundă -Diagrama Venn [9]

Istoria Învățării Profunde a început încă din 1943, s-a creat un model bazat pe rețelele neuronale ale creierului uman. Din acel moment, Învățarea Profundă a evoluat constant, au fost doar două pauze semnificative în dezvoltarea sa, întâlnite în literatura de specialitate ca iernile urâte ale Inteligenței Artificiale. În 1960 se definesc elementele de bază ale unui model continuu de propagare înapoi. [10] Deși atât de mulți algoritmi au fost dezvoltați de-a lungul timpului, nici funcțiile de activare nu au fost ignorate. Pornind de la rolul funcției de activare, adică de a filtra informațiile, dorim să avem funcții cu proprietăți la fel de specifice, întrucât se bazează pe actualizarea ponderilor din rețeaua neuronală. Pornind de la această considerație, s-au dezvoltat mai multe funcții de activare care ne permit să le folosim în diferite moduri și care ajută rețelele neuronale să atingă mai rapid convergența sau le oferă capacitatea de a utiliza mai puține straturi în definirea arhitecturii lor. Cu mai puține straturi în arhitectură, avem mai puțini parametri în rețeaua noastră deci reprezintă o bună modalitate de optimizare a rețelei. Scopul principal al funcției de activare este de a introduce neliniaritate în rezultatul unui neuron. De asemenea, funcția de activare cunoscută și sub numele de *funcție de transfer* poate decide dacă un neuron trebuie activat sau nu printr-o sumă ponderată la care se adaugă bias. O rețea neuronală fără funcții de activare este doar un model de regresie liniară. Dar funcția de activare oferă rețelei capacitatea de a învăța sarcini mai complicate.

Un alt aspect important pe care funcția de activare și determinarea inițializării ponderilor a fost dat de spațiul în care va lua valori algoritmul de optimizare. Acest algoritm vine ca o soluție pentru crearea rețelei neuronale care constă într-o problemă de optimizare non-convexă. Selectarea unei funcții de activare reprezintă un subiect important, deoarece poate afecta modul în care se schimbă datele de intrare. S-a arătat un nou tip de funcție de activare cunoscut sub numele de unitate liniară rectificată (*Rectified Linear Unit- ReLU*) care îmbunătățește performanțele rețelei neuronale din punct de vedere al eficienței în timp și a complexității

spațiale. De asemenea, s-a studiat impactul utilizării unui comportament neliniar în locul funcției sigmoide sau a funcției tangente (cunoscută și sub denumirea de *tanh*) prin utilizarea regulatorului pentru a preveni posibilele probleme numerice cu activări nelimitate. De asemenea în cadrul acestui capitol am prezentat arhitecturile *Perceptron* și *Multi-layer perceptron*.

Pentru că funcția de activare este un element fundamental în Învățarea Profundă voi menționa doar numele funcțiilor care sunt prezentare în detaliu în cadrul tezei. Aceste funcții de activare sunt:

- Funcția de activare sigmoidă
- Funcția tangentă (*tanh*)
- Funcția de activare unitate liniară rectificată ReLU (*Rectified Linear Unit*)
- Funcția de activare unitate liniară rectificată parametrică (*Parametric Rectifier - PReLU*)
- Funcția de activare *Leaky ReLU* (*LReLU*)
- Funcția de activare *unitatea liniară exponențială* (*Exponential Linear Units - ELU*)
- Funcția de activare *unități liniare scalate exponențial* (Self-Normalizing Neural Networks - SELU)
- Funcția de activare gausiană (*Gaussian Error Linear Unit – GELU*)
- Funcția de activare Swish
- Funcția de activare E-Swish
- Funcțiile de activare EliSH și HardELiSH
- Funcția de activare Flatten-T Swish (*FTS*)
- Funcția de activare Softplus
- Funcția de activare Mish
- Funcția de activare ARiA2 (*Adaptive Richard's Curve weighted Activation*)
- Funcțiile de activare SiLU și dSiLU
- Funcțiile de activare RadBas (*RadialBasis*), LogSig(*Logarithmic-Sigmoid*) și TanSig (*Tangent-Sigmoid*)
- Funcția de activare ElliotSig (*Elliot Sigmoid*)
- Funcția de activare SQNL (*Square-Law Non-Linear*)
- Funcțiile de activare ISRLU (*Inverse square root linear unit*) și ISRU
- Funcția de activare Soft Clipping(*SC*)
- Funcția de activare SReLU (*S-shaped Rectified Linear Activation Unit*)
- Funcția de activare BReLU (*Bipolar Rectified Linear Activation Unit*)
- Funcția de activare CReLU (*Concatenated Rectified Linear Unit*)
- Funcția de activare Maxout
- Funcția de activare OPLU (*Orthogonal Permutation Linear Unit Activation Functions*)
- Funcția de activare APL (***A**daptive **P**iecewise **L**inear units*)

- Funcțiile de activare Softmax, Large-margin Softmax, Noisy Softmax, SparseMax, Dropmax
- Funcția de activare Softsign
- Funcțiile de activare: KAFs, SLAF, Sinusoid, PELU (Parametric Exponential Linear Units), BLU (Bendable Linear Unit), SL-ReLU, DP ReLU și Dual Line, Hard *tanh* (*Hard Hyperbolic*), Hard sigmoid, FReLU (*flexible rectified linear unit*), Snake.

Ca strategie de învățare, algoritmul de propagare înapoi s-a dovedit a fi eficient prin faptul că asigură o clasificare a cărei precizie este în general satisfăcătoare. Principalul dezavantaj al acestei tehnici de învățare este faptul că presupune încercări repetate pentru a stabili arhitectura rețelei, numărul de straturi ascunse și numărul de neuroni în fiecare strat ascuns, în contextul în care antrenare necesită o mulțime de resurse precum memoria și timp de rulare.

Partea treia conține capitolele 3, 4 și 5.

Capitolul 3 prezintă concepte legate de tehnici de optimizare precum descendența gradientului (*Gradient Descent* - GD), descendența stocastică a gradientului (*Stochastic Gradient Descent* - SGD), evitarea supraspecializării (*overfitting*) [11], oprirea timpurie (*Early stopping*) [12], tehnica de regularizare, tehnica de *Dropout* [13], care sunt utilizate în evaluarea funcțiilor mele propuse în vederea optimizării unui model pentru a combate supraspecializarea.

Capitolul 4 cuprinde conceptele fundamentale care stau la baza înțelegerii și definirii arhitecturilor rețelelor neuronale profunde, capitol în care prezint în detaliu tipurile de arhitecturi pe care le-am folosit în cadrul capitolului 7 dedicat experimentelor. Acest capitol la rândul lui este împărțit în subcapitole care descriu succint arhitecturile folosite în cadrul tezei. Aceste arhitecturi sunt:

- Rețele neuronale de convoluție (*Convolutional Neural Network - CNN*)
- Arhitectura LeNet-5
- Rețele neuronale folosind blocuri (*Visual Geometry Group - VGG*)
- Rețea în rețea (*Network in Network - NiN*)
- Rețele reziduale (*Residual Networks - ResNet*)
- Arhitectura MobileNet
- Arhitectura Xception
- Arhitectura AlexNet

În ***capitolul 5*** îmi îndrept atenția spre o sarcină de tip *Analiză Sentiment*, folosind rețele neuronale de convoluție, pentru o sarcină de Procesare a Limbajului Natural.

Partea a patra care este partea cea mai importantă din teză conține capitolele 6 și 7.

În ***capitolul 6***, care constituie și unul dintre capitolele principale al tezei, sunt descrise funcțiile mele de activare învățabile și neînvățabile propuse pentru rețelele profunde.

După cum văzut în capitolul 2, alegerea funcției de activare are o influență covârșitoare asupra performanței rețelei neuronale. De asemenea s-a arătat că performanța este influențată inclusiv de cât de profundă este rețeaua și de inițializarea ponderilor folosind distribuția Gaussiană uniformă. În cadrul acestui capitol, voi prezenta mai multe propuneri de funcții de activare care aduc o îmbunătățire de performanță în antrenarea rețelelor neuronale.

Capitolul 6 este structurat astfel:

- În *subcapitolul 6.1. Modificarea dinamică a funcției de activare folosind algoritmul de propagare înapoi în rețelele neuronale artificiale*, propun modificarea dinamică a funcției de activare folosind o cunoscută tehnică de învățare, mai exact un algoritmul propagării înapoi (*Backpropagation-BP*). Modificarea constă în schimbarea dinamică a pantei pentru funcția de activare sigmoidă pe baza creșterii sau micșorării erorii întro epocă de învățare. Studiul a fost realizat folosind platforma WEKA (*Waikato Environment for Knowledge Analysis*) prin adăugarea acestei funcții în clasa Perceptron multistrat (*Multi-layer Perceptron -MLP*). Acest studiu urmărește modificarea dinamică a funcției de activare care s-a schimbat în funcție de eroarea relativă a gradientului și un aspect de menționat este că în definirea arhitecturii rețelei neuronale nu s-au folosit straturi ascunse pentru acest studiu. Funcțiile de activare care au fost propuse de-a lungul timpului au fost analizate prin prisma aplicabilității algoritmului BP. Scopul principal al funcției de activare constă în scalarea ieșirilor neuronilor în rețelele neuronale și introducerea unei relații neliniare între intrarea și ieșirea neuronului. Pe de altă parte, funcția sigmoidă este folosită de obicei pentru straturile ascunse, deoarece combină comportamentul liniar, curbiliniu, constant și depinde de valoarea de intrare. De asemenea, s-a demonstrat că funcția sigmoidă nu este eficientă pentru o singură unitate ascunsă, dar când sunt implicate mai multe unități ascunse, aceasta devine mai utilă. [14] Motiv pentru care în următoarele studii voi folosi arhitecturi complexe, cu mai multe straturi ascunse (rețele profunde).

  Abordarea mea în această lucrare constă în modificarea dinamică a funcției de activare folosind algoritmul BP pentru antrenarea unei rețele neuronale artificiale.

  Propun modificarea ecuației funcției sigmoide cu un parametru β care este modificat dinamic în timpul antrenamentului, cu alte cuvinte acest parametru este un parametru învățabil β (ecuația 1).

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

- În *subcapitolul 6.2. Cele mai utilizate funcții de activare: clasic versus curent*, prin acest studiu îmi propun să ofer o imagine de ansamblu sumarizată asupra celor mai utilizate funcții de activare, funcții clasice și funcții curente. Când spun clasic, mă refer la primele funcții de activare, cele mai populare și folosite în trecut. Dar, din cauza dezavantajelor lor, au apărut alte noi funcții de activare pe care le numesc curente. Aceste funcții sunt printre cele mai cunoscute funcții de activare ale Inteligenței Artificiale, ale Învățării Automate și ale Învățării Profunde. Cu fiecare funcție, ofer o scurtă descriere a funcției de activare, discut impactul acesteia și arăt domeniul unde este aplicabilă, avantajele și dezavantajele acesteia și mai multe detalii pentru o clarificare amplă. Aceste funcții acoperă mai multe probleme cum ar fi dispariția gradientului, explozia gradientului, când folosim GD și așa mai departe. Aceste soluții la aceste probleme reprezintă unul din topicurile cele mai importante din aria de cercetare și dezvoltare a Inteligenței Artificiale. În acest subcapitol am prezentat doar funcțiile prezentate în acest studiu, deoarece prezentarea în detaliu a avantajelor și dezavantajelor acestor funcții s-a făcut în cadrul capitolul 2, subcapitolul 2.3 la

prezentarea stadiului actual privind funcțiile de activare. Printre funcțiile de activare clasice am avut ca obiect de studiu: funcția de activare sigmoidă și funcția de activare tangentă hiperbolică (*tanh*). Cele două funcții sunt foarte similare, ambele necesită calcul exponențial, explozia/dispariția gradientului, diferența este dată de intervalul ieșirii activării, în cazul sigmoidei este [0,1], iar în cazul funcției tanh [-1,1]. Printre funcțiile de activare curente am avut ca obiect de studiu: ReLU, PReLU, LReLU, ELU, SELU și Softmax.

- În *subcapitolul 6.3. TeLU: o nouă funcție de activare pentru Învățarea Profundă*. Prin acest studiu îmi propun să dezvolt două funcții de activare noi care derivă din ReLU, *tanh* și funcția de activare ELU, acestea sunt destul de asemănătoare cu funcția *TanhExp [15]*, dar diferența principală constă în utilizarea funcției ELU [16] în loc de argumentul *exp* și de asemenea, proprietatea de fi învățabilă introdusă prin setarea unui parametru învățabil α. Ecuația noii funcții de activare numită TeLU învățabilă este dată de următoarea ecuație:

$$f(x) = x \cdot \tanh(elu(\alpha \cdot x)) \tag{2}$$

Unde α este un parametru învățabil, pe care l-am inițializat cu 0,1. Funcția TeLU este doar un caz particular al funcției TeLU învățabilă cu setarea parametrului la 1. Unde ELU la rândul ei poate fi scrisă astfel:

$$f(x) = elu(x) = \alpha \cdot e^{x} - 1 \tag{3}$$

Unde α> 0 (ecuația 3) este un hiperparametru care controlează valoarea la care funcția se saturează pentru intrări negative. La proiectarea funcțiilor TeLU și TeLU învățabilă ne-am inspirat din funcția TanhExp și funcția Mish. Pentru antrenare, vom avea în vedere arhitecturile ResNet18, LeNet-5, MobileNet, AlexNet, cu diferite profunzimi. Din punct de vedere al proprietăților funcțiilor de activare, TeLU și TeLU învățabilă sunt funcții centrate în 0, continue, netede, non-monotone și mărginite în partea inferioară. TeLU și TeLU învățabilă fiind două funcții mărginite inferior aduc un plus printr-o regularizare foarte puternică care conduce la o optimizare a rețelei. Pe partea pozitivă, TeLU este aproape egală cu o transformare liniară, când intrarea este mai mare de 1, valabil și pentru partea negativă când intrarea este mai mică de -1 funcția devine aproape o transformare liniară. TeLU și TeLU învățabilă arată un gradient mai abrupt aproape de zero, care poate accelera actualizarea parametrilor din rețea. Pe parcursul propagării înapoi, rețeaua își actualizează parametri, nu are probleme de oprire a antrenamentului.
- În *subcapitolul 6.4. Îmbunătățirea preciziei rețelelor neuronale profunde prin dezvoltarea de noi funcții de activare,* propun patru funcții de activare care aduc îmbunătățiri pentru seturi de date diferite în sarcinii din Vederea Artificială. Aceste funcții sunt o combinație a funcțiilor populare de activare, cum ar fi sigmoida, sigmoida bipolară, unitatea liniară rectificată (*ReLU*) și tangenta (*tanh*). Permițând funcțiilor de activare să fie învățabile, obținem modele mai robuste. Pentru validarea acestor funcții, le-am comparat cu alte funcții puternice de activare pentru a vedea cum propunerile mele au impact asupra îmbunătățirii performanței. Am folosit mai multe seturi de date,

precum setul de date Câini și Pisici, CIFAR-10, CIFAR-100. Printre arhitecturile utilizate menționăm ResNet18, ResNet34, de asemenea, am folosit tehnica Învățării prin Transfer (*Transfer Learning* -TL) în cazul utilizării arhitecturii VGG16, arhitectură pre-antrenată cu mărirea datelor (*Data Augmentation* -DA).

- Funcția de activare *TSReLU* constă în combinația dintre funcția ReLU, funcția *tanh* și funcția sigmoidă, pe care am denumit-o *TSReLU* (**T**angent-**S**igmoid-**ReLU**). Această funcție este o combinație între două funcții clasice (sigmoida și *tanh*) și o funcție curentă, dată de ReLU. Ecuația noii funcții de activare TSReLU este dată de următoarea relație:

$$f(x) = x \cdot tanh\,(sigmoid(x)) \tag{4}$$

- Funcția de activare *TSReLU învățabilă* este destul de similară cu TSReLU, singura diferență este dată de un parametru α care este învățabil. Expresia funcției TSReLU învățabilă este dată de relația:

$$f(x) = x \cdot tanh\,(\alpha \cdot sigmoid(x)) \tag{5}$$

  Atât TSReLU cât și TSReLU învățabilă sunt funcții cu curbe netede, ceea ce înseamnă că ieșirea lor va fi de asemenea lină. Această proprietate oferă o mulțime de avantaje atunci când optimizăm rețelele neuronale pentru a obține o convergență spre funcția de cost minimă.

- Funcția de activare *TBSReLU* este dată de combinația dintre funcția ReLU, funcția tangentă și funcția sigmoidă bipolară, funcție pe care am denumit-o TBSReLU (Tanh-Bipolar-Sigmoid-Relu). Expresia funcției TBSReLU este dată de relația:

$$f\left(x\right) = \mathrm{x} \cdot \tanh\left(\frac{1-e^{-x}}{1+e^{-x}}\right) \tag{6}$$

- Funcția de activare *TBSReLU învățabilă* este destul de similară cu TBSReLU, singura diferență este un parametru α care este un parametru învățabil. Expresia funcției TBSReLU învățabilă este dată de relația:

$$f\left(x\right) = \mathrm{x} \cdot \tanh\left(\alpha \cdot \frac{1-e^{-x}}{1+e^{-x}}\right) \tag{7}$$

  Funcția de activare TBSReLU învățabilă are aceleași proprietăți cu funcția de activare TBSReLU neparametrizată.

- În *subcapitolul 6.5. P-Swish: Funcție de activare cu parametri învățabili bazată pe funcția de activare Swish în Învățarea Profundă,* propun o nouă funcție de activare numită P-Swish (*Parametric Swish*), care este capabilă să aducă îmbunătățiri de performanță pe sarcini de *clasificare a obiectelor* folosind seturi de date precum CIFAR-10, CIFAR-100, dar vom vedea că am folosit și seturi de date pentru Procesarea Limbajului Natural (*Natural Language Processing NLP*). Pentru a testa această funcție nouă, am folosit mai multe tipuri de arhitecturi, printre care amintim LeNet-5, Rețea în Rețea (*Network in Network-NiN*) și ResNet34 în comparație cu funcții populare de activare, cum ar fi sigmoida, ReLU, Swish și propunerile mele. Funcția de activare a APL (*Adaptive Piecewise Linear units*) s-a dezvoltat din faptul că rețelele neuronale

artificiale au, de obicei, o funcție fixă, non-liniară de activare în fiecare neuron. APL este o nouă formă de funcție liniară de activare, care este învățată independent pentru fiecare neuron, folosind GD.

Cu toate acestea, în timp ce funcțiile de activare au fost intens explorate pentru a vedea impactul lor asupra învățării, funcția de activare pe porțiuni a fost mai puțin explorată, așa că atenția mea se concentrează pe dezvoltarea unei astfel de funcții de activare. În acest subcapitol, voi prezenta o nouă funcție de activare pe care o numesc *Parametric Swish* (P-Swish), care este o funcție de activare derivată din funcția de activare Swish. P-Swish este o funcție de activare pe porțiuni, este o combinație între funcția de activare ReLU și funcția Swish. Ecuația acestei funcții de activare poate fi definită astfel:

$$f(x) = \begin{cases} \alpha \cdot x \cdot sigmoid(\delta \cdot x) & x \leq \beta \\ x & x > \beta \end{cases} \quad (8)$$

Această funcție de activare este o funcție derivată din Swish cu 3 parametri care pot fi predefiniți sau învățabili α , β și δ atunci când x≤β, iar pentru x>β avem o funcție care derivă din funcția ReLU, dar cu parametrul β învățabil, dar în cazul când β=0 funcția devine chiar funcția ReLU. Această propunere beneficiază de proprietățile funcțiilor din care derivă Swish și ReLU, P-Swish fiind o funcție continuă, non-monotonă, nemărginită în partea superioară și mărginită în partea inferioară. Pentru a valida această funcție am folosit mai multe arhitecturi, aplicate pe sarcini variate, incluzând chiar și sarcini pentru NLP. Principalele avantaje ale funcției P-Swish sunt simplitatea și precizia îmbunătățită și faptul că nu are probleme cu dispariția gradientului, dar oferă o bună propagare a informației în timpul antrenamentului în Învățarea Profundă. În ceea ce privește această funcție de activare, am definit mai multe posibilități de reprezentare a acestei funcții prezentate în teză. Am testat funcția de activare P-Swish și pe o sarcină de segmentare semantică folosind o arhitectură de rețea complet conectată tip U-Net.

- În *subcapitolul 6.6 Noi funcții de activare bazate pe funcția de activare TanhExp în Învățarea Profundă,* propun trei funcții noi de activare (numite *TanhExp învățabilă, a_TanhExp și a_TanhExp învățabilă*), care sunt capabile să aducă îmbunătățiri de performanță pe sarcini de clasificare a obiectelor folosind seturi de date precum MNIST, Fashion-MNIST, CIFAR-10, CIFAR-100, dar vom vedea că am folosit și un set de date pentru detectarea anomaliilor în serii de timp. Pentru a le testa, am folosit mai multe tipuri de arhitecturi și le-am comparat cu funcțiile de activare ReLU, tangenta (*tanh*) și TanhExp.
    - Funcția de activare *TanhExp învățabilă*: este o nouă funcție de activare pe care am dezvoltat-o inspirându-mă din funcția de activare TanhExp. Această propunere este o funcție continuă, non-monotonă, nemărginită în partea de sus și mărginită în partea de jos. Pentru a valida această funcție am folosit mai multe arhitecturi pe care le-am mapat pe diferite tipuri de sarcini. Funcția de activare TanhExp învățabilă este destul de similară cu TanhExp, dar care aduce ca noutate parametrul său învățabil, care este capabil să aducă îmbunătățiri de precizie. Ecuația funcției TanhExp învățabilă este dată de următoarea relație:

$$f(x) = x \cdot \tanh(e^{\alpha \cdot x}) \tag{9}$$

Unde α este un un parametru care poate fi predefinit sau învățabil.

- Funcția de activare *a_TanhExp* este o funcție de activare parametrică inspirată din funcția de activare TanhExp. Parametrul *a* controlează concavitatea minimelor globale ale funcției de activare, atunci când *a* = 0 acestă funcție este chiar funcția de activare TanhExp. Variația unei scări negative reduce concavitatea și pe scara pozitivă mărește concavitatea. *a* este introdus pentru a combate scenariile de gradient "mort" datorită minimelor globale ascuțite ale funcției de activare TanhExp. Ecuația sa este definită după cum urmează:

$$f(x) = x \cdot \tanh(e^{x} + a) \tag{10}$$

- Funcția de activare *a_TanhExp* învățabilă este destul de similară cu funcția de activare a_TanhExp. În acest caz, parametrul *a* devine un parametru care poate fi învățat.

- În *subcapitolul 6.7 Dezvoltarea de noi funcții de activare în detecția anomaliilor în serii de timp cu autoencoder LSTM,* propun două funcții de activare noi în detecția anomaliilor în seriile de timp, acestea având capacitatea să reducă funcțiile de cost pe setul de validare. Pentru a atinge acest scop, am folosit un autoencoder LSTM (*Long Short-Term Memory*) [17]. Punctul cheie din propunerea mea este dat de parametrul care poate fi învățabil. Am testat propunerea mea în comparație cu alte funcții populare precum ReLU (*Linear Rectifier Unit*), tangentă hiperbolică (*tanh*) și funcția de activare TaLu. De asemenea, noutatea acestei propuneri constă în luarea în considerare a comportamentului pe porțiuni al unei funcții de activare pentru a crește performanța unei rețele neuronale în Învățarea Profundă. Propunerile mele au fost inspirate din funcția de activare TaLu (*tangent linear unit*) [18] care este o nouă funcție de activare bazată pe tangentă hiperbolică și ReLU, dezvoltată pentru rețelele neuronale și s-a dovedit că dă rezultate mai bune decât funcțiile de activare ReLU, LReLU [19] și ELU. Ecuația funcției TaLu este dată de relația:

$$f(x) = y_i = \begin{cases} x_i & \text{if } x_i \geq 0 \\ \tanh(x_i) & \text{if } \alpha < x_i < 0 \\ \tanh(\alpha) & \text{if } x_i \leq \alpha \end{cases} \tag{11}$$

Unde α este un parametru fix cu valori valori negative. A fost testat de la -0,50 până la -0,01. După cum s-a văzut, funcția TaLu este o combinație de două funcții, fiind o funcție pe porțiuni. Propunerea mea constă în dezvoltarea unor funcții noi care derivă din TaLu, dar principala diferență constă în utilizarea proprietății care le conferă noilor funcții capacitatea să fie învățabile, proprietate care lipsește în proiectarea funcției TaLu.

- Funcția de activare Talu învățabilă (*Talu learnable*) este o funcție de activare care este destul de similară cu TaLu, dar aduce ca noutate parametrul său învățabil. Ecuația funcției de activare Talu învățabilă este dată de relația:

$$f(x) = Talu \text{ learnable (x)} = y_i = \begin{cases} x_i & \text{if } x_i \geq 0 \\ \tanh(x_i) & \text{if } \alpha < x_i < 0 \\ \tanh(\alpha) & \text{if } x_i \leq \alpha \end{cases} \tag{12}$$

Unde α este un parametru învățabil.

- A doua propunere este dată de funcția de activare numită *P_Talu învățabilă* (*P_Talu learn*) care este similară cu Talu învățabilă, dar de această dată introducem doi parametri care pot fi învățabili. Ecuația sa este definită după cum urmează:

$$f(x) = P\_Talu(\mathrm{x}) = y_i = \begin{cases} x_i & \text{if } x_i \geq b \\ \tanh(x_i) & \text{if } a < x_i < b \\ \tanh(a) & \text{if } x_i \leq a \end{cases} \quad (13)$$

Unde a și b sunt doi parametri învățabili. Abordarea mea s-a bazat pe strategii de ultimă generație, care constau în funcții de activare cu un singur parametru scalabil care se învață în timpul antrenamentului. [20]

- În *subcapitolul 6.8 Soft Clipping Mish - o nouă funcție de activare*, propun două funcții de activare compoziționale: *Soft Clipping Mish* (SC Mish) și *Soft Clipping Mish învățabilă* (Soft Clipping Mish learnable – SCL Mish), testate pe o sarcină de predicție a poluării aerului pe date tip serii de timp multivariate. Pentru a modela această problemă am folosit două arhitecturi: LSTM (*Long Short Term Memory*) și GRU (*Gated Recurrent Unit*), oferind astfel un studiu comparativ între rezultate prin prisma funcției de cost și a RMSE (*root mean square error*). Punctul cheie și în această propunere este dat de parametrul învățabil.
  - *Soft Clipping Mish* (SC Mish) este o funcție de activare care derivă din Mish, fiind dată de ecuația:

$$f(x) = max(0, x \cdot tanh(softplus(x))) \quad (14)$$

  - În plus pornind de la această propunere, am propus și *Soft Clipping Mish învățabilă* (Soft Clipping Mish learnable – SCL Mish) care practic derivă din Soft Clipping Mish, diferența majoră constând în parametrul α învățabil, oferind mai multă flexibilitate rețelei.

$$f(x) = max(0, x \cdot tanh(softplus(\alpha \cdot x))) \quad (15)$$

Unde α este un parametru învățabil.

Ultima parte a lucrării conține capitolele 7 și 8, respectiv bibliografia.

În ***capitolul 7*** prezint rezultatele și analiza rețelelor profunde cu funcțiile propuse de activare pe seturile de date alese pe le-am analizat din punct de vedere teoretic în capitolul 6.

Capitolul 8 conține concluziile tezei, direcții de cercetare viitoare pe marginea acestei teme. Lucrarea se încheie cu publicațiile prezentate în cadrul conferințelor internaționale și publicate pe durata pregătirii doctorale. Totodată, în teză sunt prezentate **11** lucrări publicate de către autorul tezei ca prim-autor. Succint, după nivelul de impact, gruparea lucrărilor publicate este următoarea:

- 1 lucrare în revistă indexată Web of Science (ISI) – Factor de Impact 1.324.
- 7 lucrări în volume ale unor manifestări științifice (proceedings) indexate Web of Science (ISI).
- 2 lucrări în volumele unor manifestări științifice indexate BDI (IEEE Xplore) – care vor fi indexate în Web of Science (ISI).
- 3 lucrări în volumele unor manifestări științifice din care 3 vor fi indexate în Web of

Science (ISI).

Teza are 243 de pagini, din care: 205 pagini structurate în 8 capitole, 12 pagini de bibliografie și 16 de pagini dedicate anexelor. Lucrarea conține 176 figuri și 301 de titluri bibliografice.

Contribuțiile mele aduse în această teză prin dezvoltarea de funcții de activare noi, evidențiate în mod comparativ cu funcțiile de activare existente sunt următoarele:

(i) Adaptarea funcției sigmoide.

(ii) Cele mai utilizate funcții de activare: clasic versus curent.

(iii) TeLU: o nouă funcție de activare pentru Învățarea Profundă.

(iv) Îmbunătățirea preciziei rețelelor neuronale profunde prin dezvoltarea de noi funcții de activare: TSReLU, TSReLU învățabilă, TBSReLU și TBSReLU învățabilă.

(v) P-Swish: Funcție de activare cu parametri învățabili bazată pe funcția de activare Swish în Învățarea Profundă.

(vi) Noi funcții de activare bazate pe funcția de activare TanhExp în Învățarea Profundă: TanhExp învățabilă, a_TanhExp și a_TanhExp învățabilă.

(vii) Dezvoltarea de noi funcții de activare în detecția anomaliilor în serii de timp univariate cu autoencoder LSTM: Talu învățabilă și P_Talu învățabilă.

(viii) Soft Clipping Mish - o nouă funcție de activare, atât cu parametru predefinit cât și cu parametru învățabil.

Până în prezent, am următoarele lucrări acceptate și prezentate la conferințele internaționale respectiv jurnal internațional din domeniul Automaticii și Calculatoarelor:

1. *Marina Adriana Mercioni*, Ștefan Holban, "*The recognition of the architectural style using Data Mining techniques*", Conferința: 12th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 17-19 mai 2018, Pg: 331-337 Publicată: 2018, indexată ISI, Scopus.
2. *Marina Adriana Mercioni*, Ștefan Holban, "*Evaluating hierarchical and non-hierarchical grouping for develop a smart system*", Conferința: 13th International Symposium on Electronics and Telecommunications (ISETC), Timisoara, Romania, 8-9 noiembrie 2018, Pg: 114-117 Publicată: 2018, indexată ISI, Scopus.
3. *Marina Adriana Mercioni*, Alexandru Tiron, Ștefan Holban, "*Dynamic Modification of Activation Function using the Backpropagation Algorithm in the Artificial Neural Networks*", Jurnal: International Journal of Advanced Computer Science and Aplications, Volum: 10 Issue: 4 Pg: 51-56 Publicată: Aprilie 2019, indexată ISI, Scopus.
4. *Marina Adriana Mercioni*, Nina Holban, Vlad Virgiliu Todea, "*Wireless Routers and their impact on the environment*", Global and Regional in Environmental Protection, Conferența GLOREP 2018, 15- 17 Noiembrie 2018, Timisoara, Romania.
5. *Marina Adriana Mercioni*, Ștefan Holban, "*A survey of distance metrics in clustering data mining techniques*", ICGSP '19 Proceedings of the 2019 3rd International Conference on Graphics and Signal Processing, Pages 44-47, Hong Kong, Publicată: 01 – 03 Iunie 2019, indexată Scopus.
6. *Marina Adriana Mercioni*, Ștefan Holban, "*A study on hierarchical clustering and the distance metrics for identifying architectural styles*", 9 th International Conference on Energy and Environment 2019, 17-18 Octombrie 2019, Timisoara Romania.
7. *Marina Adriana Mercioni*, Ștefan Holban, "*The Most Used Activation Functions:*

Classic Versus Current", 15th International Conference on Development and Application Systems, Suceava, Romania, May 21-23, 2020.

8. *Marina Adriana Mercioni*, Angel Marcel, Tat, Ștefan Holban, "*Improving the Accuracy of Deep Neural Networks Through Developing New Activation Functions*", 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP 2020), Cluj-Napoca, Romania, September 3-5, 2020.
9. *Marina Adriana Mercioni*, Ștefan Holban, "*Novel Activation Functions Based on TanhExp Activation Function in Deep Learning*", 2020 International Conference on Data Science, Machine Learning and its Applications (ICDML 2020), Sridevi Women's Engineering College (SWEC), New Delhi, India, October 9-10, 2020.
10. *Marina Adriana Mercioni*, Ștefan Holban, "*P-Swish: Activation Function With Learnable Parameters Based on Swish Activation Function in Deep Learning*", International Symposium on Electronics and Telecommunications 2020, Timisoara, Romania, November 05 - 06 2020.
11. *Marina Adriana Mercioni*, Ștefan Holban, "*TeLU: A New Activation Function for Deep Learning*", International Symposium on Electronics and Telecommunications 2020, Timisoara, Romania, November 05 - 06 2020.
12. *Marina Adriana Mercioni*, Ștefan Holban, „*Soft Clipping Mish – A Novel Activation Function for Deep Learning*", The 4th International Conference on Information and Computer Technologies (ICICT 2021), Kahului, Maui Island, Hawaii, United States, March 11-14, 2021.
13. *Marina Adriana Mercioni*, Ștefan Holban, „*Developing Novel Activation Functions in Time Series Anomaly Detection with LSTM Autoencoder*", IEEE 15th International Symposium on Applied Computational Intelligence and Informatics, May 19-21, 2021.

Bibliografie selectivă

[1] A. Krizhevsky et al., *ImageNet Classification with Deep Convolutional Neural Networks*, 2012.
[2] G. Koch et al., *Siamese Neural Networks for One-shot Image Recognition*, 2015.
[3] A. Turing, *Computing Machinery and Intelligence*, 1950.
[4] Zimek, Arthur; Schubert, Erich, *Outlier Detection*, 2017.
[5] A. Panigrahi et al., A. Panigrahi et al., *Effect of activation functions on the training of overparametriezed neural nets*, 2020.
[6] Y. LeCun, Y. Bengio, and G. Hinton, *Deep learning*, 2015.
[7] V. Nair et al., *Rectified linear units improve restricted boltzmann machines*, 2010.
[8] P. Ramachandran et al., *Searching for Activation Functions*, 2017.
[9] C. E. Nwankpa et al., *Activation Functions: Comparison of Trends in Practice and Research for Deep Learning*, 2018.
[10] H. J. Kelley, *Gradient theory of optimal flight paths*, 1960.
[11] P. Bühlmann et al., *Statistics for High-Dimensional Data*, 2011.
[12] L. Prechelt et al., *Early Stopping — But When?*, 2012.
[13] G. E. Hinton, *System and method for addressing overfitting in a neural network*, 2016.
[14] K. Hara et al., *Comparison of activation functions in multilayer neural network for pattern classification*, 1994.
[15] Xinyu Liu et al., *TanhExp: A Smooth Activation Function with High Convergence Speed for Lightweight Neural Networks*, 2020.
[16] D. A. Clevert et al., *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)*, 2015.
[17] A. Pulver et al., *LSTM with Working Memory*, 2017.

[18] M. Jain, *A New Hyperbolic Tangent Based Activation Function for Neural Networks*, 2018.
[19] K. He et al., *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*, 2015.
[20] A. Turner et al., Julian Francis, *Neuroevolution: Evolving heterogeneous artificial neural networks*, 2014.