UNIVERSITATEA BABEȘ-BOLYAI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

Optimizarea Arhitecturilor Rețelelor Neuronale folosind Metode din Domeniul Inteligenței Computaționale

Rezumatul tezei de doctorat

Student doctorand: Sergiu Cosmin Nistor
Conducător științific: Prof. dr. Czibula Gabriela

2021

Cuvinte cheie: Neural architecture search, Graph neural networks, Rețele neuronale recurente, Rețele neuronale convoluționale.

Cuprins

Cuprinsul tezei de doctorat

Lista publicațiilor

Clasamentul publicațiilor a fost realizat conform standardelor CNATDCU (Consiliul Național de Atestare a Titlurilor, Diplomelor și Certificatelor Universitare) aplicabile pentru studenții doctoranzi înmatriculați după 1 octombrie 2018. Toate clasamentele sunt listate conform clasificării jurnalelor[1] și a conferințelor[2] în Informatică.

Publicații indexate în Web of Science - Science Citation Index Expanded

[NC22] **Sergiu Cosmin Nistor** and Gabriela Czibula *IntelliSwAS: Optimizing Deep Neural Network Architectures using a Particle Swarm-based Approach*. Expert systems with Applications, Volume 187, January 2022, 115945 (**2020 IF=6.954**, 2020 Journal IF Quartile Q1)

Rank A, 8 points.

[NMM+21] **Sergiu Cosmin Nistor**, Mircea Moca, Darie Moldovan, Delia Beatrice Oprean and Răzvan Liviu Nistor. *Building a Twitter Sentiment Analysis System with Recurrent Neural Networks*. Sensors (2021), 21, 7, 2266, MDPI (**2020 IF=3.576**, 2020 Journal IF Quartile Q1).

Rank A, 2.66 points.

[NMN21] **Sergiu Cosmin Nistor**, Mircea Moca and Răzvan Liviu Nistor. *Discovering Novel Memory Cell Designs for Sentiment Analysis on Tweets*. Genetic Programming and Evolvable Machines (2021): 22, 2, 147-187, Springer (**2020 IF=1.714**, 2020 Journal IF Quartile Q2).

Rank B, 4 points.

[NID20] **Sergiu Cosmin Nistor**, Tudor Alexandru Ileni and Adrian Sergiu Darabant. *Automatic Development of Deep Learning Architectures for Image Segmentation*. Sustainability (2020), 12, 22, 9707, MDPI (**2020 IF=3.251**, 2020 Journal IF Quartile Q2).

Rank B, 4 points.

Publicații indexate în Web of Science, Conference Proceedings Citation Index

[NDB18] **Sergiu Cosmin Nistor**, Adrian Sergiu Darabant and Diana Borza. *Micro-Expressions Detection Based on Micro-Motions Dense Optical Flows*. 2018 26th International Conference

[1] https://uefiscdi.ro/premierea-rezultatelor-cercetarii-articole
[2] http://portal.core.edu.au/conf-ranks/

on Software, Telecommunications and Computer Networks (SoftCOM), pp. 1-7. IEEE, 2018 (**indexed IEEE**).

Rank B - CORE2018, 4 points.

[BND17] Diana Borza, **Sergiu Cosmin Nistor** and Adrian Sergiu Darabant. *Towards Automatic Skin Tone Classification in Facial Images*. International Conference on Image Analysis and Processing (ICIAP), pp. 299-309. Springer, Cham, 2017.

Rank B - CORE2017, 4 points.

[NMDB17] **Sergiu Cosmin Nistor**, Alexandra-Cristina Marina, Adrian Sergiu Darabant and Diana Borza. *Automatic gender recognition for "in the wild" facial images using convolutional neural networks*. 2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), pp. 287-291. IEEE, 2017.

Rank C - CORE2017, 1 point.

[Nis20] **Sergiu Cosmin Nistor**. *Multi-Staged Training of Deep Neural Networks for Micro-Expression Recognition*. 2020 IEEE 14th International Symposium on Applied Computational Intelligence and Informatics (SACI), pp. 000029-000034. IEEE, 2020.

Rank D - CORE2020, 1 point.

[Nis21] **Sergiu Cosmin Nistor**. *An Actor-Critic Approach to Neural Network Architecture Search for Facial Expressions Recognition*. 2021 17th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), IEEE, 2021, accepted for publication.

Rank D - CORE2021, 1 point.

Scorul publicațiilor: 29.66 puncte.

Introducere

Progresul rapid al informaticii ne-a permis să automatizăm numeroare procese. Au fost propuse atât de multe aplicații încât există foarte puține domenii care încă nu au beneficiat de acest progres. Ne sunt oferite două perspective importante: ceea ce este generat de procesele care au fost automatizate deja și ce poate fi automatizat în viitor. Prima perspectivă se referă la datele care sunt generate. Aceste date pot fi analizate, iar rezultatele pot fi folosite pentru a îmbunătăți calitatea procesului automatizat sau pentru a dezvolta noi aplicații.

Chiar dacă acest progres există, mai există de asemenea multe oportunități de dezvoltare. Momentan, implicarea experților este necesară în numeroase domenii. Învățarea automată (ML) a ajutat în dezvoltarea soluțiilor bazate pe cantitățile mari de date care există. Deși multe aplicații sunt bazate pe astfel de metode inteligente, aproape toate aplicațiile necesită studierea și dezvoltarea algoritmilor, ceea ce poate fi realizat doar de către experții umani.

Pentru fiecare nouă aplicație, experții aleg, de obicei, o arhitectură existentă care s-a dovedit deja eficientă în rezolvarea altor probleme. Apoi, această arhitectură este adaptată noului caz de utilizare. Arhitectura este antrenată și testată, în scopul evaluării calității ei. Rezultatele sunt analizate de experți, iar be baza acestor analize sunt făcute ajustări arhitecturii. Noul candidat este evaluat. Acest proces repetitiv este necesar pentru descoperirea unei soluții eficiente. De asemenea, acest proces necesită implicarea experților care trebuie să facă analiza riguroasă și să ia deciziile importante.

Comunitatea științifică a dezvoltat un interes pentru automatizarea procesului de găsire a arhitecturii optime de rețea neuronală pentru rezolvarea unei anumite probleme și astfel a apărut domeniul *neural architecture search* (NAS). Acești algoritmi automatizează cât de mult este posibil din procesul de căutare a arhitecturilor. Aceștia au fost aplicați atât pentru rețelele neuronale convoluționale (CNN), cât și pentru rețelele neuronale recurente (RNN). În cazul ideal, o asemenea soluție ar obține rapid arhitectura optimă pentru rezolvarea unei anumite probleme, fără intervenție umană. Deși acest domeniu are un potențial de aplicabilitate foarte mare și rezultatele care au fost obținute deja sunt impresionante, mai există încă multe provocări care trebuie depășite.

Experiențele noastre anterioare au implicat propunerea de soluții în care extrăgeam anumite caracteristici din date folosind reguli stabilite de noi, nu învățate de algoritmi. De asemenea, a fost nevoie să experimentăm cu numeroase arhitecturi pe care le-am studiat, comparat și ajustat manual. Aceste abordări clasice pentru proiectele de ML s-au dovedit, într-adevăr, de succes în multe situații. Chiar și așa, procesul ar putea fi îmbunătățit prin automatizare. Multe arhitecturi dezvoltate manual au fost propuse, iar apoi reutilizate în multe alte aplicații. Deși alegerea unor astfel de arhitecturi are șanse mari de a conduce către soluții de succes, poate fi și o limitare pentru că lasă multe opțiuni neexplorate.

În această teză, propunem propriile noastre tipuri de algoritmi NAS. Soluțiile noastre pot fi folosite atât pentru propunerea arhitecturilor pentru CNN, cât și RNN. Pentru a demonstra calitatea algoritmilor noștri, i-am aplicat la rezolvarea a două probleme provocatoare, dar și cu grad mare de aplicabilitate: clasificare de imagini și analiză de sentiment pe tweet-uri.

Contribuții originale

Împărțim prezentarea contribuțiilor noastre originale bazându-ne pe tipul de rețele neuronale pentru care căutăm arhitecturi eficiente: CNN și RNN. Metodele pe care le propunem au fost construite astfel încât să beneficieze de particularitățile acestor tipuri de rețele.

Capitolul 2 este dedicat RNN. Am prezentat o serie de metode pe care le-am folosit pentru găsirea arhitecturilor eficiente în scopul rezolvării unei anumite probleme. Problema selectată este analiza de sentiment pe tweet-uri, o problemă provocatoare care poate fi modelată ca o problemă de procesare de secvențe, ceea ce o face abordabilă pentru RNN.

Prima abordare pe care am folosit-o este una experimentală care urmează metodologia clasică de căutare a arhitecturilor, metodologie care a fost folosită de multe ori cu succes. Această abordare necesită o implicare considerabilă din partea cercetătorilor de-a lungul întregului proces de căutare. Rezultatele acestei abordări sunt publicate în [NMM+21]. Aici am definit modul în care modelăm problema și am experimentat cu diferite decizii de proiectare. Am explorat diferite opțiuni de preprocesare și decizii arhitecturale. Prezentăm aceste experimente și rezultatele obținute. Aceste rezultate au fost folosite în dezvoltarea celorlalte abordări mai sofisticate.

A doua abordare este un algoritm original NAS, abordare pe care am publicat-o în [NMN21]. Am propus un algoritm evolutiv (EA) a cărui scop este descoperirea unor noi celule de memorie recurente. Algoritmul, împreună cu componentele sale, este descris în detaliu. Am propus strategii de estimare a performanței în scopul reducerii timpului necesar pentru identificarea celulelor de înaltă calitate. A fost descoperit un număr mare de celule, dintre care le prezentăm pe cele mai bune, pe care le comparăm cu celulele cu scop generic existente deja în literatura de specialitate.

A treia abordare dedicată RNN se concentrează pe strategia de estimare a performanței. Deși am propus în contribuția noastră originală anterioară asemenea strategii, acestea încă aveau nevoie de timpi lungi de execuție. Noua strategie a fost să folosim un algoritm ML pentru estimarea performanței. Propunem un nou model pentru a prezice cât de bine își va îndeplini scopul o celulă de memorie recurentă, bazându-ne pe structura acesteia. Au fost efectuate experimente ample pentru a găsi algoritmul care să facă preziceri de performanță cu mare acuratețe. Folosind algoritmii descoperiți, am analizat noi celule de memorie recurente și am prezentat-o pe cea mai bună dintre acestea.

Capitolul 3 prezintă abordarea noastră pentru descoperirea unur noi arhitecturi CNN. Pentru acest tip de rețele am decis să alegem clasificarea de imagini ca problemă pentru care să căutăm arhitecturile. Această contribuție originală a fost publicată în [NC22]. Metoda pe care o propunem se numește *IntelliSwAS* și este o abordare de tip *particle swarm* pe care o extindem cu un algoritm ML pentru estimarea rapidă a performanței.

Am propus un algoritm de tip *Particle Swarm Optimization* (PSO) pentru căutarea de celule CNN care pot fi folosite pentru a construi arhitecturi întregi. Acest algoritm este configurabil pentru că am dorit să putem găsi o strategie de căutare eficientă. Particulele se mișcă prin spațiul de căutare care reprezintă celulele CNN, pentru care am propus o reprezentare numerică. Am creat un mediu de simulare în care să căutăm configurațiile care să producă aritecturi de înaltă performanță. În acest mediu de simulare am rulat algoritmii noștri PSO cu diferite opțiuni pentru parametri și am evaluat rezultatele.

Particulele trebuie să știe care sunt pozițiile de înaltă calitate și este de dorit ca acest lucru să se facă cu cât mai puține resurse de calcul. În acest scop, am propus un nou algoritm ML special conceput pentru prelucrarea datelor structurate ca grafuri orientate aciclice (DAG). Folosim acest algoritm pentru a prezice calitatea relativă a arhitecturilor candidate. Am efectuat experimente pentru crearea acestui algoritm și am comparat, de asemenea, eficiența acestuia cu un model bazat pe lucrări

care au fost propuse anterior în literatura științifică.

Configurațiile parametrilor PSO și modelul de estimare a performanței au fost folosite pentru a descoperi noi celule CNN eficiente. Aceste celule au fost folosite pentru a construi arhitecturi CNN complete și au fost efectuate experimente pentru a evalua calitatea acestor arhitecturi.

Structura tezei

Primul capitol al acestei teze este dedicat lucrărilor care ne-au inspirat în cercetarea noastră. Ne bazăm pe aceste lucrări și, din acest motiv, am decis să oferim o privire de ansamblu asupra multiplelor subiecte care ne-au influențat. Deși această revizuire nu este exhaustivă și multe contribuții științifice importante nu au fost incluse, sperăm să dăm o idee despre elementele de bază pe care le-am folosit, progresele realizate și provocările care există.

Capitolul 2 prezintă contribuțiile noastre originale a căror scop este descoperirea arhitecturilor RNN.

În capitolul 3 am prezentat contribuțiile noastre originale care au vizat căutarea automată a arhitecturilor CNN.

Concluziile le tragem în ultimul capitol. Am creat o imagine de ansamblu asupra a ceea ce am prezentat și am sintetizat rezultatele obținute. Am identificat în acest capitol și posibile direcții viitoare de cercetare pe care ne propunem să le dezvoltăm.

Capitolul 1

Fundamente teoretice

Acest capitol este dedicat prezentării contribuțiilor științifice care au fost relevante pentru cercetarea noastră. În primul rând, două dintre secțiuni sunt dedicate celor doi algoritmi ML pentru care ne propunem să găsim arhitecturi eficiente: RNN și CNN. Mai mult, soluțiile pe care le propunem folosesc și RNN pentru căutare, așa că este important să detaliem modul în care funcționează și care sunt punctele lor forte. Oferim apoi o imagine detaliată a evoluției algoritmilor NAS pentru a crea contextul în care ne-am efectuat cercetările și am propus propriile versiuni ale unor astfel de algoritmi. Deoarece rețelele neuronale pot fi reprezentate cu ușurință sub formă de grafuri, am considerat că studierea *graph neural networks* (GNN) ne poate oferi oportunități importante de procesare a arhitecturilor. În capitolele ulterioare vom prezenta propriile versiuni de GNN pe care le-am propus pentru îmbunătățirea algoritmilor NAS dezvoltați. O secțiune este dedicată analizei sentimentelor pe tweet-uri, deoarece aceasta este problema pe care am ales-o pentru soluțiile noastre de căutare a arhitecturii pentru RNN. Pentru arhitecturile CNN, am selectat clasificarea imaginilor, deoarece aceasta este o problemă importantă în domeniul viziunii computerizate și este strâns legată de evoluția CNN, așa cum menționăm și în secțiunea dedicată acestui tip de rețele.

Capitolul 2

Metode pentru Căutarea Arhitecturilor de Rețele Neuronale Recurente

În acest capitol am prezentat abordările noastre pentru a descoperi noi arhitecturi RNN. Am propus mai multe soluții, fiecare bazându-se pe cea precedentă. Pentru aceste soluții, am selectat ca problemă analiza sentimentului pe tweet-uri. Această problemă este potrivită, fiind o problemă de procesare de secvențe, pentru care RNN este o alegere adecvată ca algoritm și este, de asemenea, o problemă relevantă, primind mult interes în comunitatea științifică datorită aplicațiilor sale importante. Chiar dacă am selectat această problemă, metodele noastre sunt aplicabile la multe alte probleme cu nevoi minime de modificări, așa cum va fi explicat în secțiunile următoare.

Am publicat aceste abordări în două articole originale: [NMM+21, NMN21]

Prima abordare constă în căutarea în maniera clasică de a crea arhitecturi, evaluând calitatea acestora prin antrenarea până la convergență, testarea lor și utilizarea rezultatelor pe care le obținem pentru a propune noi arhitecturi. Pentru aceasta, am analizat arhitectura dintr-o perspectivă macro, utilizând două celule de memorie RNN existente și ajustând mai multe decizii de proiectare care includ modul de utilizare al acestora într-o arhitectură RNN completă. Am propus și prezentat anterior această abordare în [NMM+21]. Această fază ne-a permis să evaluăm performanța diferitelor decizii arhitecturale. Am experimentat cu mai multe arhitecturi bazate pe conceptele prezentate în Capitolul 1. Soluția noastră convertește mai întâi textul într-o reprezentare numerică. Apoi, o preprocesare este aplicată pe datele de intrare. Informațiile sunt transmise unuia sau mai multor straturi recurente pentru procesare. Opțional, se aplică apoi un mecanism de atenție. Clasificarea finală se face printr-un strat de tip *feed-forward*, care emite scorurile pentru fiecare sentiment considerat, pozitiv sau negativ, scorul mai mare fiind cel care decide clasa. Textul introdus este procesat caracter cu caracter și nicio parte a textului tweet-ului nu este filtrată. Această decizie a fost luată pentru a ține cont de toate informațiile disponibile la analiza tweet-ului. Alte lucrări [KWM11, PP10] folosesc un dicționar de cuvinte. Prin procesarea numai a caracterele alfanumerice și prin gruparea acestora în cuvinte, se pierd informații precum emoticoanele, emoji-urile și hashtag-urile. Chiar dacă aceste caracteristici sunt tratate separat, prin utilizarea unui dicționar de cuvinte predefinit, cuvintele scrise în mod creativ s-ar pierde pentru că nu se află în dicționar sau ar trebui corectate, ceea ce ar putea introduce erori în text și distorsiona rezultatul.

A doua abordare este una de tip NAS. Am dezvoltat un EA pe care îl folosim pentru căutarea arhitecturilor RNN. Am publicat această soluție în [NMN21]. Pentru această abordare, am analizat arhitecturile dintr-o perspectivă micro, căutând noi celule de memorie RNN. Arhitecturile complete sunt construite pe baza acestor celule și a șabloanelor pe care le-am creat pe baza informațiilor obținute din căutarea clasică pe care am efectuat-o în abordarea anterioară. Metoda noastră este un EA

complet cu toate componentele, care sunt descrise în detaliu. Pentru funcția de aptitudine, evaluăm calitatea arhitecturilor RNN folosind mai multe tehnici, dar nu ML. Folosind rezultatele obținute în abordarea anterioară, am selectat șablonul de arhitectură pentru crearea acesteia la nivel macro și am decis să schimbăm arhitectura unităților ascunse, propunând alternative la celulele cu scop generic LSTM [HS97] și GRU [CVMG+14]. Am folosit trei probleme diferite pentru a descoperi și a evalua modelele. Am efectuat experimente și rezultatele arată că cele mai bune modele obținute depășesc termenii de comparație aleși — care sunt cele mai populare celule, LSTM și GRU. În timpul procesului de căutare, am evaluat aproximativ 17000 de celule. Candidatul câștigător pe care l-am selectat i-a depășit pe ceilalți pentru problema analizei sentimentelor în general, arătând, prin urmare, generalitate. Am făcut selecția câștigătorului utilizând acuratețea cumulată pentru toate cele trei probleme luate în considerare.

Algoritmul folosit în soluția noastră este un EA. Acest tip de algoritm simulează evoluția speciilor. Se folosește o populație de indivizi, fiecare individ reprezentând o posibilă soluție a problemei.

În spațiul nostru de căutare, un individ reprezintă o celulă de memorie RNN candidată. Structura fiecărui individ este dată de genele acestuia. Pentru a forma noi soluții, genele indivizilor sunt recombinate și apoi mutate. Indivizii sunt clasificați în funcție de aptitudine, iar cei cu performanțe de top sunt favorizați pentru transmiterea genelor. Componentele EA sunt:

- Reprezentarea individului;
- Operatorul de selecție;
- Operatorul de recombinare;
- Operatorul de mutație;
- Inițializarea populației;
- Evaluarea aptitudinii.

A treia noastră abordare este centrată pe o strategie mai sofisticată de estimare a performanței bazată pe un algoritm ML. Din nou, căutăm celule de memorie RNN, dar accelerăm procesul propunând un nou algoritm pentru prezicerea calității candidaților noștri, algoritm care rulează considerabil mai rapid decât metodele anterioare pe care le-am folosit în acest scop. Experimentând cu abordarea anterioară, am observat că evaluarea performanței unei arhitecturi candidate necesită o cantitate mare de timp, chiar și atunci când se utilizează strategiile de estimare a performanței pe care le-am propus. Pentru a reduce acest timp, propunem un algoritm ML dedicat procesării grafurilor pe care îl folosim pentru estimarea performanței unei celule de memorie recurente pentru o anumită problemă. Am prezentat arhitectura algoritmului de estimare și am analizat fiecare componentă. Am inclus acest nou algoritm într-o metodă completă NAS pe care am descris-o în detaliu. Folosind acest algoritm, am putut evalua un milion de arhitecturi de celule de memorie recurente și am descoperit modele noi care obțin performanțe bune în analiza sentimentelor. Am descris arhitectura descoperită care a condus la obținerea celor mai bune performanțe.

Algoritmul propus, pe care l-am denumit *Graph-Encoding Recurrent Neural Network* (GERNN), are ca date de intrare grafuri care sunt compuse dintr-un set de noduri și un set de muchii între aceste noduri. Fiecare nod poate avea oricâte proprietăți, dar toate nodurile trebuie să aibă același set de proprietăți. În mod similar, muchiile pot avea propriul set de proprietăți. Grafurile pot fi orientate sau neorientate. Rezultatul GERNN depinde de problema care trebuie rezolvată, deoarece algoritmul nostru poate fi folosit atât pentru probleme de clasificare, cât și pentru probleme de regresie.

GERNN este un algoritm compus din rețele recurente și mecanisme de atenție și din acest motiv, datele de intrare trebuie să aibă reprezentări numerice. Nodurile și muchiile sunt convertite în vectori caracteristici care descriu proprietățile asociate nodului / muchiei. Reprezentarea numerică a grafului este o listă de vectori-nod și vectori-muchie. Această reprezentare secvențială este impusă și de RNN care compun algoritmul, deoarece RNN sunt concepute pentru procesarea secvențelor [LBE15].

După preprocesare, nodurile sunt procesate de o RNN. O altă RNN este folosită pentru a procesa muchiile. Între cele două RNN există un mecanism de atenție care este utilizat pentru a selecta mai bine care sunt cele mai importante rezultate ale procesării nodurilor pentru procesarea muchiilor. Alte două mecanisme de atenție sunt utilizate pentru a agrega rezultatele procesării nodurilor și rezultatele procesării muchiilor într-un vector caracteristic pe baza căruia se face predicția finală.

Prezentările din acest capitol se bazează pe lucrările originale pe care le-am publicat [NMM+21, NMN21].

Capitolul 3

Metode pentru Căutarea Arhitecturilor de Rețele Neuronale Convoluționale

În acest capitol am prezentat o metodă originală de căutare automată a arhitecturilor CNN folosind o abordare de tip PSO. Am publicat lucrarea în [NC22].

Contribuția acestei lucrări este dublă. În primul rând, propunem un nou model ML conceput pentru a procesa arhitecturile CNN și a estima performanța acestora. Modelul propus extinde RNN astfel încât să fie mai potrivite pentru procesarea datelor structurate ca DAG [FSZ14], nu doar date secvențiale. DAG sunt grafuri cu muchii orientate (datele sunt transferate într-o singură direcție) și fără cicluri în interiorul grafului (niciun vârf nu poate fi un fiu al său).

Deoarece arhitecturile CNN au o astfel de structură DAG, unitățile de calcul fiind reprezentate ca vârfuri și fluxul de date reprezentat ca muchii orientate, modelul nostru poate fi utilizat pentru a le analiza și a face predicții în mod eficient. Folosim acest model pentru a accelera căutarea. Numim abordarea noastră *Intelligent Swarm Architecture Search* (*IntelliSwAS*), deoarece folosim PSO pentru căutarea arhitecturilor CNN, dar ne îmbunătățim algoritmul cu modelul ML. Modul în care folosim modelul nostru pentru estimarea performanței este, de asemenea, nou. Nu folosim abordarea obișnuită de estimare a performanței unei singure arhitecturi date, ci luăm în considerare perechi de arhitecturi și prezicem calitatea relativă a celor doi candidați.

Pentru a doua contribuție a acestei lucrări, am efectuat experimente folosind abordarea noastră *IntelliSwAS* și am evaluat performanța arhitecturilor CNN rezultate. Au fost efectuate comparații cu metodele de referință pentru clasificarea imaginilor și analize statistice și am observat și raportat că soluția noastră depășește multe alte opțiuni, arătând că abordarea propusă oferă o îmbunătățire semnificativă.

În timpul conceperii și implementării acestei abordări am fost ghidați de următoarele întrebări de cercetare:

RQ1 Cum pot fi extinse modelele RNN astfel încât acestea să fie mai potrivite pentru procesarea datelor structurate ca DAG?

RQ2 Ar putea un astfel de model ML orientat spre DAG să fie utilizat pentru a prezice performanța unei arhitecturi CNN complete pe baza structurii blocului principal al acestei CNN?

RQ3 În ce măsură poate acest model ML, în combinație cu o abordare PSO, să fie folosit pentru a descoperi arhitecturi CNN de înaltă performanță pentru clasificarea imaginilor?

RQ4 Îmbunătățește *IntelliSwAS* semnificativ performanța clasificării imaginilor în comparație cu lucrările de referință existente?

Pentru a răspunde la RQ1, am propus o extensie la RNN pe care am descris-o în detaliu. Modelul propus este folosit pentru a răspunde la RQ2. Abordarea noastră este diferită de cea obișnuită în algoritmii NAS. În timp ce alte soluții estimează care ar putea fi calitatea unei singure arhitecturi, noi propunem un model care compară două arhitecturi candidate. În acest scop, experimentăm cu estimarea calității relative a diferitelor arhitecturi CNN și evaluăm capacitatea modelului nostru de a rezolva această problemă. Am prezentat aceste experimente și, de asemenea, o comparație a modelului cu o alternativă existentă. În legătură cu RQ3, am propus algoritmul *IntelliSwAS*. Se efectuează apoi o analiză statistică pentru a răspunde la RQ4 și a evidenția performanța *IntelliSwAS* în raport cu modelele existente de clasificare a imaginilor. Am descoperit noi arhitecturi CNN pe care le-am descris.

Algoritmul nostru caută celule CNN. Sunt construite arhitecturi CNN complete prin înlănțuirea mai multor instanțe ale aceleiași celule. Un șablon fix descrie modul în care sunt organizate instanțele celulei.

Fiecare celulă are un număr fix de noduri de calcul. Fiecare nod preia datele de intrare de la alte noduri, le concatenează pe dimensiunea adâncimii și aplică o operație asupra acestor date pentru a-și produce rezultatele. Nodurile de calcul și, deci, celulele, iau ca date de intrare un volum de dimensiunea $H \times W \times C$ și produc un volum de ieșire de dimensiunea $H \times W \times C'$. După cum se poate observa, celulele nu afectează înălțimea și lățimea datelor, dar numărul de canale poate fi modificat, în funcție de numărul de filtre. Chiar dacă nodurile sunt unități de calcul independente, am decis că toate nodurile unei celule vor avea același număr de filtre pentru omogenitate. Fiecare nod are asociată o operație pe care o aplică.

Celulele pot fi reprezentate ca DAG. Fiecare celulă are un nod de intrare, un nod de ieșire și noduri intermediare. Nodurile respectă o ordine și, pentru a păstra proprietatea DAG (fără cicluri), fiecare nod poate avea conexiuni numai la nodurile care sunt înaintea lui. Așadar, primul nod intermediar poate fi conectat numai la nodul de intrare, al doilea nod intermediar poate fi conectat la nodul de intrare și la primul nod intermediar și așa mai departe.

O celulă poate fi reprezentată prin matricea ei de adiacență și o listă de identificatori de operație (câte unul pentru fiecare nod). Deoarece nodul de intrare nu are nevoie de descriere, putem ignora linia sa din matricea de adiacență. De asemenea, niciun nod al celulei curente nu va lua ca date de intrare valoarea de la nodul de ieșire, astfel coloana pentru acest nod poate fi ignorată. Considerăm 8 operații posibile pentru nodurile noastre, așadar fiecare identificator de operație poate fi reprezentat pe 3 biți. Deoarece am impus ca nodurile să primească date de intrare numai de la nodurile anterioare lor în ordonare, matricea de adiacență va avea doar zerouri deasupra diagonalei principale. În acest fel, putem reprezenta o celulă ca o secvență de biți, care poate fi convertită într-un număr zecimal.

Am căutat cea mai bună arhitectură CNN folosind PSO. Algoritmul nostru este ajustabil printr-un set de parametri. O populație de particule caută printre posibilele celule de dimensiune N. Fiecare rulare a algoritmului folosește o dimensiune constantă a populației, care este numărul de particule care vor căuta celule CNN eficiente. Fiecare particulă are o poziție, care este un număr întreg și călătorește prin spațiul de căutare cu o viteză. Fiecare poziție codifică o celulă CNN și conversia se face așa cum am explicat anterior. N este un parametru al căutării noastre și rulăm căutările PSO pentru diferite valori ale acestui parametru.

Am propus *Directed Acyclic Graph Recurrent Neural Network* (DAGRNN), răspunsul nostru la **RQ1**. În cazul general, modelul nostru ML este folosit pentru a procesa date structurate ca DAG. În cazul particular al *IntelliSwAS*, folosim DAGRNN ca strategie de estimare a performanței. Majoritatea strategiilor de estimare a performanței prezentate în literatura științifică procesează o singură arhitectură. Această arhitectură este luată ca date de intrare și performanța sa estimată constituie rezultatul. Abordarea noastră este diferită. Noi luăm două arhitecturi și folosim modelul nostru pentru

a estima calitatea relativă a acestora.

Avem nevoie de un algoritm ML capabil să compare performanțele a două arhitecturi CNN fără antrenarea și testarea acestor CNN. Algoritmul nostru își fundamentează predicția pe structura CNN. Cele două CNN care trebuie comparate se bazează pe două celule CNN. Am creat o variație a RNN pentru a reprezenta și a procesa mai bine structura de tip DAG a celulelor.

Modelul DAGRNN va procesa graful nod cu nod. În timp ce RNN iau în considerare starea internă la pasul anterior, DAGRNN iau în considerare stările interne la toate nodurile care au o muchie care se conectează la nodul procesat la momentul curent. Deoarece avem nevoie ca aceste stări interne să fie disponibile atunci când procesăm nodul curent, DAGRNN folosește o ordonare topologică. Chiar dacă pot exista mai multe ordonări topologice pentru același DAG, calculele efectuate de DAGRNN sunt aceleași pentru fiecare astfel de ordonare.

Prezentarea din acest capitol se bazează pe lucrările originale pe care le-am publicat [NC22, NID20, Nis21].

Concluzii și Direcții Viitoare de Cercetare

Algoritmii inteligenți ne-au ajutat să automatizăm multe procese într-o mare varietate de domenii. Chiar dacă au fost găsite atât de multe aplicații pentru astfel de algoritmi, aceștia au încă un potențial mare de a rezolva multe alte probleme. Provocarea este de a proiecta algoritmul care poate rezolva eficient problema. Pentru aceasta, abordarea clasică necesită implicarea unor experți umani pentru propunerea și evaluarea continuă a arhitecturilor de rețele neuronale candidate până la obținerea unui rezultat satisfăcător.

În această teză de doctorat ne-am concentrat pe domeniul NAS, care are ca scop automatizarea propunerii unui algoritm eficient pentru o anumită problemă. Prin utilizarea rețelelor neuronale artificiale, NAS experimentează automat cu mai multe arhitecturi candidate în scopul de a o găsi pe cea mai potrivită.

Am prezentat multiple soluții originale pe care le-am propus în acest domeniu. Soluțiile propuse au fost grupate în două categorii principale în funcție de tipul de rețele neuronale pe care le-am căutat: RNN și CNN. Chiar dacă unele componente ale unui algoritm NAS pot fi adaptate pentru a fi utilizate pentru ambele tipuri de algoritmi, aceste rețele neuronale au particularități diferite pe care am dorit să le luăm în considerare în soluțiile noastre.

Pentru RNN, am prezentat trei abordări. Deoarece aveam nevoie de o problemă la care să lucrăm, am selectat analiza sentimentelor pe tweet-uri.

Prima abordare este un studiu experimental în care am experimentat cu mai multe decizii arhitecturale pentru construirea unei RNN pentru analiza sentimentelor pe tweet-uri. Am ajuns la concluzia că cea mai bună preprocesare este, de departe, utilizarea vectorilor *one-hot*. Am observat că nu există nicio diferență semnificativă în rezultate atunci când se utilizează LSTM sau GRU, cele două cele mai populare celule de memorie. Mecanismele de atenție pe care le-am folosit la nivel de rețea nu au adus îmbunătățiri metodei noastre. De asemenea, am observat care au fost numărul de straturi și numărul de unități care au produs cele mai bune performanțe.

Folosind informațiile obținute din prima abordare, am propus un nou algoritm NAS pentru căutarea de noi celule de memorie care să producă RNN eficiente pentru analiza sentimentelor pe tweet-uri. Soluția propusă este un EA pe care l-am proiectat special pentru a funcționa cu celule de memorie RNN. Pentru a asigura generalitatea soluțiilor, am folosit trei probleme pe care am testat celulele de memorie. Pe lângă cea bazată pe un set mare de date, am folosit și alte două probleme bazate pe un set de date mai mic pentru a accelera căutarea, dar și pentru a vedea dacă celulele descoperite pentru o problemă ar funcționa bine pe alta. Am obținut modele foarte performante, egalând și chiar depășind modelele de referință care au folosit celulele clasice LSTM sau GRU. Am evaluat aproximativ 17000 de modele, dintre care le-am selectat pe cele mai performante și le-am analizat în detaliu.

Pentru a treia abordare, am propus o strategie mai sofisticată de estimare a performanței. Soluția pe care am propus-o este un nou model GNN, pe care l-am denumit GERNN. L-am definit și folosit pentru estimarea calității celulelor de memorie recurente. Pe baza setului de date de perechi de celule RNN și a acurateței corespunzătoare pe care le-am construit, am antrenat mai multe arhitecturi GERNN. GERNN au fost comparate pe baza rezultatelor și au fost analizate diferitele decizii de

proiectare. Le-am folosit pe cele mai performante pentru a căuta celule RNN noi.

Folosind GERNN, am evaluat un milion de arhitecturi noi pe care le-am generat, am descoperit noi celule de memorie recurente și am descris-o și prezentat-o pe cea care obține cele mai bune performanțe, pe care am denumit-o ERMC.

Chiar dacă celula nou descoperită nu a depășit alternativele existente pe care le-am considerat, a obținut rezultate similare, demonstrând că metodologia poate duce la noi modele de celule RNN de succes. GERNN s-a dovedit a ajuta la găsirea rapidă de noi modele, dar în acest moment l-am asociat doar cu căutarea aleatorie. Combinarea algoritmului de estimare a performanței cu o strategie de căutare mai complexă ar trebui să producă rezultate și mai bune.

Folosind experimentele efectuate, am demonstrat că GERNN este capabil să estimeze cu precizie calitatea unei RNN pe o anumită problemă pe baza structurii celulei care este utilizată. Mai mult, GERNN este capabil să facă această estimare mult mai rapid decât alte strategii de estimare a performanței, ajutând la îmbunătățirea timpilor de rulare a algoritmilor de căutare a arhitecturii neuronale. GERNN este suficient de generic pentru a face predicții pentru RNN pentru orice problemă și suficient de generic pentru a fi inclus în alți algoritmi NAS.

Al doilea tip de rețele neuronale spre care ne-am concentrat atenția a fost CNN. Am propus, descris și analizat *IntelliSwAS*, o metodă de descoperire a arhitecturilor CNN eficiente pentru clasificarea imaginilor. Pentru căutare, am folosit PSO. Particulele noastre explorează spațiul de căutare care conține celule CNN candidate și comunică între ele pentru a putea converge către soluții eficiente.

Am căutat cele mai bune configurații pentru versiunea noastră a algoritmului PSO și am analizat cele mai bune configurații rezultate. Apoi, am folosit aceste configurații pentru a efectua căutarea propriu-zisă a celulelor. Dintre celulele descoperite, le-am luat în considerare doar pe acelea care au avut cele mai bune performanțe pentru dimensiunile lor respective. Aceste celule de înaltă calitate au fost apoi integrate într-o arhitectură CNN mai mare pe care am testat-o cu mare succes pe mai multe seturi de date de clasificare a imaginilor. Am efectuat analize statistice pentru a demonstra că rezultatele obținute îmbunătățesc cu adevărat rezultatele raportate în literatura științifică.

Direcții viitoare de cercetare

Activitatea de cercetare poate fi extinsă în mai multe direcții. *IntelliSwAS* s-a dovedit a fi eficient în găsirea de celule CNN de înaltă calitate, totuși aceste celule trebuiau integrate manual în arhitecturi CNN mai mari. În mod similar, șablonul de arhitectură pe care l-am folosit pentru RNN a fost selectat prin căutare manuală. Aceasta este o limitare a metodelor noastre, care este dată de spațiul de căutare exponențial care ar trebui explorat pentru a descoperi arhitecturi complete CNN sau RNN. Intenționăm să explorăm în continuare diverse metode care ar putea extinde soluțiile noastre la căutarea de arhitecturi complete și să găsim rezultate de înaltă calitate într-un timp rezonabil.

În legătură cu cele două noi tipuri de GNN pe care le-am propus, GERNN și DAGRNN, acestea ar putea fi integrate în algoritmi NAS diferiți sau chiar utilizate împreună. DAGRNN profită de structura DAG a CNN, dar nu există niciun motiv pentru a nu experimenta și cu GERNN în acest context. DAGRNN ar trebui teoretic să aibă rezultate mai bune, dar astfel de experimente ar fi într-adevăr interesante. Utilizarea DAGRNN pentru RNN nu se poate face în forma sa actuală, deoarece RNN nu sunt DAG, dar ar putea fi creată o posibilă extensie în acest scop.

Deoarece deocamdată am folosit GERNN doar împreună cu căutarea aleatorie, o posibilă direcție viitoare este includerea acestuia într-un algoritm NAS cu o strategie de căutare mai sofisticată.

Modelele GNN pe care le-am propus ar putea fi folosite și în alte scopuri decât pentru estimarea performanței. Le-am putea include și în strategia de căutare. O posibilă idee în această direcție ar fi

înlocuirea operatorului de recombinare din EA cu o metodă bazată pe GERNN care i-ar lua pe cei doi părinți și i-ar recombina automat într-un descendent.

În contextul extinderii utilizării modelelor GNN, am putea chiar explora aplicații în alte domenii decât NAS. Există multe date structurate sub formă de grafuri, iar unele aplicații care procesează astfel de date ar putea beneficia de pe urma GERNN și DAGRNN.

Bibliografie

[BND17] Diana Borza, Sergiu Cosmin Nistor, and Adrian Sergiu Darabant. Towards automatic skin tone classification in facial images. In *International Conference on Image Analysis and Processing*, pages 299–309. Springer, 2017.

[CVMG+14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[FSZ14] Ronja Foraita, Jacob Spallek, and Hajo Zeeb. *Directed Acyclic Graphs*, pages 1481–1517. Springer New York, New York, NY, 2014.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[KWM11] Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. Twitter sentiment analysis: The good the bad and the omg! In *Fifth International AAAI conference on weblogs and social media*, 2011.

[LBE15] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[NC22] Sergiu Cosmin Nistor and Gabriela Czibula. IntelliSwAS: Optimizing deep neural network architectures using a particle swarm-based approach. *Expert Systems with Applications*, 187:115945, 2022.

[NDB18] Sergiu Cosmin Nistor, Adrian Sergiu Darabant, and Diana Borza. Micro-expressions detection based on micro-motions dense optical flows. In *2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–7. IEEE, 2018.

[NID20] Sergiu Cosmin Nistor, Tudor Alexandru Ileni, and Adrian Sergiu Dărăbant. Automatic development of deep learning architectures for image segmentation. *Sustainability*, 12(22):9707, nov 2020.

[Nis20] Sergiu Cosmin Nistor. Multi-staged training of deep neural networks for micro-expression recognition. In *2020 IEEE 14th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pages 000029–000034. IEEE, 2020.

[Nis21] Sergiu Cosmin Nistor. An actor-critic approach to neural network architecture search for facial expressions recognition. In *20121 17th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE, 2021.

[NMDB17] Sergiu Cosmin Nistor, Alexandra-Cristina Marina, Adrian Sergiu Darabant, and Diana Borza. Automatic gender recognition for "in the wild" facial images using convolutional neural networks. In *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 287–291. IEEE, 2017.

[NMM+21] Sergiu Cosmin Nistor, Mircea Moca, Darie Moldovan, Delia Beatrice Oprean, and Răzvan Liviu Nistor. Building a twitter sentiment analysis system with recurrent neural networks. *Sensors*, 21(7):2266, 2021.

[NMN21] Sergiu Cosmin Nistor, Mircea Moca, and Răzvan Liviu Nistor. Discovering novel memory cell designs for sentiment analysis on tweets. *Genetic Programming and Evolvable Machines*, 22(2):147–187, Jun 2021.

[PP10] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326, 2010.