



UNIVERSITATEA DE VEST DIN TIMIȘOARA
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
PROGRAMUL DE STUDII DE MASTER:
Inginerie software și Securitate cibernetică

Traffic Sign Recognition

Supervizor:

Lect. Dr. Mădălina Erașcu

Studenti:

Andreea Bîra-Negrut

Bogdan Pinghireac

Eduard Marian Neguriță

Larisa Drăgănescu

Abstract

Acest studiu explorează performanța și robustețea rețelelor neuronale în recunoașterea semnelor de circulație. În detalierea rezultatelor experimentale, raportul abordează, de asemenea, instrumentele utilizate și provocările întâmpinate în procesul lor de instalare.

Contents

| | | |
|----------|----------------------------|----------|
| 1 | Introducere | 4 |
| 2 | Dataset | 4 |
| 3 | Instalare Tool-uri | 4 |
| 3.1 | Alpha Beta Crown | 4 |
| 3.1.1 | Introducere | 4 |
| 3.1.2 | Instalare | 5 |
| 3.2 | Marabou | 6 |
| 3.2.1 | Introducere | 6 |
| 3.2.2 | Instalare | 6 |
| 4 | Rulare benchmark | 6 |
| 4.1 | Alpha Beta Crown | 6 |
| 4.2 | Marabou | 7 |
| 5 | Bibliografie | 7 |

1 Introducere

Rețelele neuronale profunde au adus o transformare semnificativă în modul în care interacționăm cu tehnologia în viața de zi cu zi. Acestea au devenit un factor determinant în domenii precum mașinile autonome, smartphone-uri, jocuri, drone etc., marcând o schimbare fundamentală în modul în care sistemele complexe sunt create și utilizate. Capacitatea lor de a învăța modele complexe din seturi de date și de a face prognoze sau decizii în funcție de acestea le face o unealtă valoroasă în domeniile menționate și în multe altele.

Algoritmii Alpha-Beta-Crown și Marabou reprezintă instrumente și structuri de verificare folosite în contextul evaluării rețelelor neuronale. Acești algoritmi au scopul de a furniza analize detaliate ale modului în care rețelele neuronale se comportă și de a aborda provocările asociate testării acestora.

2 Dataset

Traffic Signs Recognition[2][3][4] este folosit pentru a automatiza recunoașterea semnelor de circulație și este deosebit de important în sistemele avansate de asistență. Aceasta reprezintă o problemă complexă în domeniul viziunii artificiale și recunoașterii modelelor.

Setul de date conține peste mii de imagini care reprezintă diferite semne de circulație. Aceste imagini reflectă variații semnificative în aspectul vizual al semnelor, din cauza factorilor precum distanța, iluminarea, condițiile meteorologice, obstrucții parțiale și unghiurile de rotație. Fiecare imagine este însoțită de mai multe seturi de caracteristici pre-calculate, ceea ce facilitează utilizarea algoritmilor de învățare automată fără a fi nevoie de cunoștințe avansate în prelucrarea imaginilor.

Setul de date cuprinde 43 de clase, a căror frecvențe nu sunt echilibrate. Participanții la benchmark au de clasificat două seturi de testare, fiecare conținând peste 12.000 de imagini.

3 Instalare Tool-uri

3.1 Alpha Beta Crown

GitHub: <https://github.com/Verified-Intelligence/alpha-beta-CROWN>

3.1.1 Introducere

Alpha Beta Crown sau α, β -Crown[5] este un tool care verifica o rețea neuronală construit pe un cadru care utilizează propagarea liniară legată și tehnici de ramificare legate.

Tool-ul este foarte eficient dacă este folosit cu un GPU deoarece are mai multe resurse decât un CPU și astfel execuția este mult mai rapidă. De asemenea se poate scala în mod eficient la rețele convoluționale de dimensiuni semnificative, inclusiv cele cu milioane de parametri.

Este compatibil cu o varietate extinsă de arhitecturi de rețele neuronale, cum ar fi CNN, ResNet și diferite funcții de activare, datorită bibliotecii versatile auto_LiRPA[6]

dezvoltate de aceeași echipă.

auto_LiRPA reprezintă o bibliotecă pentru derivarea și calcularea automată a limitelor folosind analiza perturbațiilor bazată pe relaxare liniară (LiRPA) pentru rețelele neuronale, fiind o unealtă utilă pentru verificarea formală a robusteții.

Algoritmul LiRPA folosește de asemenea un algoritm de graf pe grafuri computaționale generale definite prin intermediu PyTorch.

α, β -Crown este castigatorul în competițiile VNN-COMP (International Verification of Neural Networks Competition), din anul 2021, 2022 și 2023, obținând cel mai bun scor total și depășind semnificativ alte tool-uri de verificare a rețelelor neuronale pe o gamă extinsă de peste 2 ani.

3.1.2 Instalare

Pentru instalare avem nevoie de un sistem de operare Linux (noi am ales WLS prezent în Windows) și următoarele programe instalate, altfel se pot genera erori:

- git
- unzip
- nvidia driver
- cuda tool-kit
- anaconda sau miniconda conform documentației alpha-beta-CROWN

Primul pas este clonarea repository folosind următoarele comenzi:

```
git clone git@github.com:Verified-Intelligence/alpha-beta-CROWN.git
cd alpha-beta-CROWN
git clone git@github.com:Verified-Intelligence/auto_LiRPA.git
```

Dupa clonare trebuie creat environment-ul general folosind următoarele comenzi

```
conda env create -f complete_verifier/environment.yaml --name alpha-beta-crown
```

```
conda activate alpha-beta-crown
```

Dupa acest pas se rulează următoarele comenzi:

```
cd auto_LiRPA
python setup.py install
```

Dupa terminarea instalării am încercat un exemplu prezent în repository și anume cifar_resnet_2b.yaml.

```
cd complete_verifier
python abcrown.py --config exp_configs/tutorial_examples/cifar_resnet_2b.yaml
```

Dupa testarea unui exemplu am trecut la instalarea benchmark-ului[1] folosind următoarele comenzi:

```
git ../..
git clone git@github.com:ChristopherBrix/vnncomp2023_benchmarks.git
cd vnncomp2023_benchmarks
sh setup.sh
```

Dupa acest pas se poate trece la rularea benchmark-ului.

```

BaB round 15
batch: 1120
Average branched neurons at iteration 15: 1.0000
splitting decisions:
split level 0: [/25, 949] [/25, 949] [/25, 940] [/25, 949] [/28, 372] [/19, 52] [/25, 572] [/28, 372] [/25, 572]
pruning_in_iteration open status: True
ratio of positive domain = 786 / 2240 = 0.3508928571428571
pruning-in-iteration extra time: 0.02131342887878418
Time: prepare 1.0932 bound 1.0953 transfer 0.0091 finalize 0.4021 func 2.5999
Accumulated time: func 15.0670 prepare 4.4163 bound 8.7559 transfer 0.0586 finalize 1.7663
Current worst splitting domains lb-rhs (depth):
-0.13904 (23), -0.13604 (23), -0.13601 (23), -0.13433 (23), -0.13363 (23), -0.13287 (23), -0.13264 (23), -0.13191 (23), -0.13126 (23),
-0.12629 (23), -0.12507 (23), -0.12390 (23), -0.12290 (23), -0.12144 (23), -0.12131 (23),
length of domains: 1454
Time: pickout 0.0078 decision 1.8045 set_bounds 0.8594 solve 2.6013 add 0.0388
Accumulated time: pickout 0.0744 decision 8.5311 set_bounds 3.6271 solve 15.0734 add 0.3502
Current (lb-rhs): -0.1390395164489746
6327 domains visited
Cumulative time: 27.725351095199585

```

Figure 1:

3.2 Marabou

3.2.1 Introducere

to be done

3.2.2 Instalare

to be done

4 Rulare benchmark

4.1 Alpha Beta Crown

Dupa finalizarea instalarii tool-urilor avem instalat tot mediul de lucru pentru a putea rula tool-ul `alpha_beta_crown`.

Pentru rulare putem opta pentru rulare prin GPU sau prin CPU, noi am ales GPU pentru ca are putere de calcul mai mare. (pentru rulare prin CPU se adauga urmatorul argument **-device CPU** la comanda de executie)

Comenzile pentru rulare sunt urmatoarele:

```
cd alpha-beta-CROWN/complete_verifier
```

```
python abcrown.py --config exp configs/vnncomp23/gtrsb.yam
```


- [2] Kaoutar Sefrioui Boujemaa, Ismail Berrada, Afaf Bouhoute, Karim Boubouh. “Traffic sign recognition using convolutional neural networks”. In: *IEEEExplore* (2017). URL: <https://ieeexplore.ieee.org/abstract/document/8238205>.
- [3] Lai Jia Shyan, T H Lim, Dk Norhafizah Pg Hj Muhammad. “Real time road traffic sign detection and recognition systems using Convolution Neural Network on a GPU platform”. In: *IEEEExplore* (2023). URL: <https://ieeexplore.ieee.org/document/10007277>.
- [4] Twine AI. “The Best Road Sign Detection Datasets of 2022”. In: *twine* (2022). URL: <https://www.twine.net/blog/road-sign-detection-datasets/>.
- [5] Verified-Intelligence. *Verified-Intelligence: alpha-beta-CROWN*. URL: <https://github.com/Verified-Intelligence/alpha-beta-CROWN>. (accessed: 15.01.2024).
- [6] Verified-Intelligence. *Verified-Intelligence: auto_LiRPA*. URL: https://github.com/Verified-Intelligence/auto_LiRPA. (accessed: 15.01.2024).