UNIVERSITATEA DE VEST DIN TIMIŞOARA
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ

Verification of Binarized Neural Networks using alpha-beta-CROWN and Marabou

Rafael-Valentin Ban
Cosmin-Ştefan Negureanu
Mihai-Iosif Fârţală
Cristina-Larisa Petcu
Mădălina-Maria Radu

Conf. Dr. Mădălina Eraşcu

Abstract

Contents

Chapter 1

Introduction

Chapter 2

Dataset description

The experiment we decided to tackle is the one presented in the paper **Architecturing Binarized Neural Networks for Traffic Sign Recognition** [9]. The aforementioned experiment deals with **binarized neural networks** for traffic sign recognition. The paper discusses a bottom-up approach for designing BBNs by studying their constituent layer characteristics, like binarized convolutional layers, max pooling, batch normalization and fully connected layers. The authors of this paper study these aspects by exploring various combinations of the aforementioned layers with different kernel sizes, numerous filters and neurons using the **German Traffic Sign Recognition Benchmark** as training data.

According to the paper, the **German Traffic Sign Recognition Benchmark** (**GTSRB**) is a multi-class single-image dataset that is used in the paper's experiment as training data for the BNNs. It consists of images of road signs of different types such as prohibitory signs ("No Entry", "Wrong Way", "Road Closed", "No Straight Ahead" and so on), danger signs ("Under Construction", "Do Not Cross", "Train Track", "Slippery Road" and so on), as well as mandatory signs ("Go Straight Ahead", "Turn Right", "Permitted Directions", "Buses / Trucks / Bicycles / Pedestrians Only" and more). The images in the dataset are of different sizes, with a range from 25 x 25 to 243 x 225, with some not being perfect squares. As a total count, it contains 39,209 images used for training and 12,630 used for the actual experiment.

The diversity of the dataset does not end at the different classes of traffic signs: each sign has multiple photos presenting it in different shades of the same color, different colors, color degradation, lighting conditions, perspective change, shade as well as integrity and degradation. These numerous differences make the dataset more versatile, allowing the model to better identify each sign in multiple varied situations. However, the variety of the photos makes it a challenge for the model to accurately identify each type of sign; sometimes it can be challenging even for a human.

Besides the main GTSRB dataset, the authors also made use of additional traffic signs data; more specifically, they also used traffic signs from the **Belgian Traffic Sign Database** as well as the **Chinese Traffic Sign Database**. The Belgian dataset contains roughly 7095 of 62 different classes; the Chinese dataset contains 5998 traffic sign images of 58 different classes. However, despite the addition of supplementary data, only a subset of the classes in these two aforementioned dataset actually match the

GTSRB dataset. As a consequence, the authors preprocessed the images, eliminating the images that did not match the standard German Traffic Signs presented in the main dataset; and relabeling those that did appear in the German database as well. In the end, 1818 from the Belgian Traffic Signs dataset and 1590 from the Chinese dataset were used in the experiment.

In terms of our experiment, we used the entire German Traffic Sign Recognition Benchmark dataset as well as the additional traffic signs data from the Belgian Traffic Signs and Chinese Traffic Signs databases. Our main purpose was to improve the initial experiment and fix the problems that the authors encountered.

Figure 2.1: Some images used in the German Traffic Signs Recognition Benchmark

Chapter 3

Tools

3.1 alpha-beta-CROWN

alpha-beta-CROWN is a neural network verifier based on an efficient linear bound propagation framework and branch and bound, it can provide provable robustness guarantees against adversarial attacks and can also verify other general properties of neural networks[1].

For the installation of the alpha-beta-CROWN, we have used the step by step "Instructions for running the VNN-COMP benchmarks"[2] published by the tool developers which contains everything required for a successful installation. Firstly we cloned the git project recursively which also cloned auto_LiRPA[3] to also support a wide range of neural network architectures.

After having the project locally, we installed Miniconda[4] to be able to have a fresh environment in which we can install the dependencies and run the tool on our benchmark. Here we ran into our first problem, which appeared to be that when trying to create the environment with Miniconda[4], the yaml extension file defined to specify the dependencies when creating the environment, was containing a redirection to another yaml extension file. The specified redirection did not work and we had to move the dependencies from the redirected yaml extension file to the main yaml extension file.

Once we successfully used the correct yaml file for environment configuration, we ran into our second problem, which was the fact that not all of the dependencies used for alpha-beta-CROWN are available for Windows, therefore we had to change the operating system to a Linux based one and try again.

Finally, when we retried all of the steps with the adaptations written above, the installation was successful and the environment was ready for running benchmarks.

A problem that we encountered on the journey after the installation, was that each time we started the environment, we had to redefine the system variable VNNCOMP_PYTHON_PATH which contains the path to python, otherwise the run was not possible to start.

3.2 Marabou

Chapter 4

Experimental results

When it comes on running the benchmark, we have reached the conclusion that the first benchmark, tllverifybench[5], was too simple and flawless to be able to extend the project after a simple run that reproduced the exact output from the competition. Therefore we chose to change it to Traffic Signs Recognition[6] so we can try to solve some of the problems that the competition run had, where we achieved the following values:

#	Tool	Verified	Falsified	Penalty
1	alpha-beta-CROWN	0	39	-3
2	Marabou			

4.1 alpha-beta-CROWN

After the first run we observed that due to a timeout after an instance that timed out, we could not reproduce the exact output due to the process stopping at that point, generating the report only to that specific instance (see

results_traffic_signs_recognition_default_time.csv[7]). After increasing the timeout in run_all_categories.sh file found in the alpha-beta-CROWN project, we were able to reproduce the output, but the instance that crashed the process still timed out, this time after 1000.0 seconds (the given value, see

results_traffic_signs_recognition_increased_time_to_1000.csv[8]).

The falsified responses comes from the fact that, the output for most of the instances, seem to be satisfiable when in fact there is still generated an counter-example for 39 of the instances which turn to be a false positive. The penalty comes from 3 instances that have the result different than "sat" or "unsat". Two of them have the result "no_result_in_file" and after having a look in the logs, it appears that the branch-and-bound (BaB) round 69 stops (kills) the process in both instances therefore there is no output file from which to get the result. No successful method in changing the BaB and timeout configurations was found yet after some failed attempts.

The remaining instance that fails with the "run_instance_timeout" seems to not be fixed after increasing the timeout from 480 seconds to 1000, further timeout increase will be made to test if any real timeout value can fix the problem.

4.2 Marabou

Chapter 5

Conclusions

In conclusion, it seems that the dataset Traffic Signs Recognition[6] it is very hard to be verified 100% because even if a small adaptation rate (epsilon) like 1.00, the image will differ unless we set an adaptation rate of 0. Beside of this, both of the tools found penalties for this dataset for some verification cases which had a small adaptation rate number (1.00 and 5.00) which indicates that for those cases the image could not be verified unless we increase the specified rate.

Bibliography

[1] Alpha-beta-crown git repository, "https://github.com/Verified-Intelligence/alpha-beta-CROWN"

[2] Alpha-beta-CROWN instructions for running the VNN-COMP benchmarks, "https://github.com/Verified-Intelligence/alpha-beta-CROWN/blob/main/complete_verifier/docs/vnn_comp.md"

[3] Auto-LiRPA repository, "https://github.com/Verified-Intelligence/auto_LiRPA"

[4] Miniconda, "https://docs.conda.io/projects/miniconda/en/latest/"

[5] TLLVerifyBench repository, "https://github.com/jferlez/TLLVerifyBench"

[6] Traffic Signs Recognition Repository, "https://github.com/ChristopherBrix/vnncomp2023_bench

[7] results_traffic_signs_recognition_default_time.csv "https://github.com/RafaelBan/VFProject/blo

[8] results_traffic_signs_recognition_increased_time_to_1000.csv "https://github.com/RafaelBan/VFProject/blob/main/results_traffic_signs_recognition_increase

[9] Architecturing Binarized Neural Networks for Traffic Sign Recognition "https://arxiv.org/abs/2303.15005"