

Repeatability Package

This folder contains a repeatability package for:

- Paper: Set-Based Training for Neural Network Verification
- Venue: TLMR

Folder Structure

- `./` : base path
 - `./code` : path to code
 - * `./cora` : path to CORA
 - * `./scripts` : path to auxiliary scripts
 - * `./main.m` : **main Matlab script**
 - `./data` : path to data
 - `./results/<evalname>` : path to results (created after execution)
 - * `./evaluation` : path to store any evaluation results
 - * `./plots` : path to plots
 - * `./results.txt` : logs of all outputs to command window
 - `./Dockerfile` : Dockerfile
 - `./license.lic` : place license file here
 - `./README.md` : read me file (this file)
 - `./run.sh` : **main script** to run from command line
 - `./settings.sh` : settings for scripts
 - `./screen.sh` : script to run `run.sh` within a linux screen

Step 1: Installation

This folder contains the code as well as a docker file to run the code in one click (see below).

However, you need to provide a Matlab license. You can specify i) a license server, ii) a license file, or iii) run it directly in Matlab:

Option 1: License server (recommended)

- Ask your Matlab administrator if a Matlab license server is available.
- In `settings.sh`, configure the license server : `LICENSE_SERVER=<port>@<hostname>`.
- Proceed with Step 2: Run from command line

Option 2: License file

Download a license file `license.lic` to run the code:

- Create a Matlab license file: For the docker container to run Matlab, one has to create a new license file for the container. Log in with your Matlab account at <https://www.mathworks.com/licensecenter/licenses/>. Click on your license, and then navigate to
 1. “Install and Activate”
 2. “View activated computers”
 3. “Activate a Computer” (... may differ depending on how your licensing is set up).
- Choose:
 - Release: R2024b
 - Operating System: Linux
 - Host ID: 0242AC11000a (= Default MAC of docker container)
 - Computer Login Name: matlab
 - Activation Label: <any name>
- When prompted if the software is already installed, choose “Yes”.
- Download the file and place it next to **Dockerfile**.
- Proceed with Step 2: Run from command line.

Step 2: Run the code

Run from command line (recommended)

You can run the evaluation in one click in a docker container using the `run.sh` script (see bug fix: windows/linux line breaks below):

```
./run.sh <evalname> <gpu-device>
```

where the argument `<evalname>` is used to name the evaluation run (defaults to `datetime`), and the optional argument `<gpu-device>` is used to select the GPU (see GPU settings below).

Use `{mnist,mnist-ablation,cifar10,svhn,tinyimagenet}` for the argument `<evalname>` to evaluate corresponding trained models and produce a table containing the results.

The results will be stored to `./results/<evalname>` after the evaluation finishes. To view intermediate results during the evaluation, you can copy the current `results` folder out of the docker container using

```
docker cp "$DOCKER_NAME":/results .
```

where `DOCKER_NAME` is as in `settings.sh` or using `docker ps`.

If you are using linux screens, you can also call

```
./screen.sh <evalname> <gpu-device>
```

which might be helpful when running the script on a server to ensure it finishes correctly even if your connection is interrupted. You can always detach from the screen using `CTRL+A+D` and reattach using

```
screen -rd $SCREEN_NAME
```

where `SCREEN_NAME` is as in `settings.sh` or using `screen -ls`.

Evaluate Trained Models

Training all models used in the evaluation takes a long time. We can provide the trained models upon request. The directory `results` contains the models trained for our ablation studies, i.e., by setting the argument `<evalname>` to `mnist-ablation` the trained models are evaluated and a table containing the results is produced. The trained models for `{mnist,cifar10,svhn,tinyimagenet}` can be provided upon request.

Training Models

Setting the argument `<evalname>` to a different name than the directories containing the trained models will train models based on the specified configuration in `main.m`. For your convenience, by default a small 3-layer convolutional neural network is trained on MNIST and evaluated (takes ~30min); e.g., run the command:

```
./run.sh mnist-short <gpu-device>
```

Disclaimer: The models are only trained for 10 epochs, thus their accuracies are lower compared to the reported accuracies in the paper.

By default only the MNIST dataset is provided. Other datasets can be downloaded and included in the `datasets` directory.

GPU Settings

For docker to use the GPU, you have to specify the `<gpu-device>` docker should use. You can find your available GPUs using the command `nvidia-smi`. Possible options are the GPU id (e.g., 0), `all`, and `none` (default). Read more about it here: <https://docs.docker.com/desktop/features/gpu/>.

Please note that this setting might not be necessary for this repeatability package.

Run from Matlab

Alternatively, open this directory in Matlab and run:

```
addpath(genpath('./code'));  
main('<evalname>');
```

where the optional argument `<evalname>` is used to name the evaluation run (defaults to `datetime`). The results will be stored to `./results/<evalname>`.

Note: Please ensure that all required toolboxes for CORA are installed (see Step 1: Option 3 above).

Important Notes

- When running the evaluation in docker, docker might randomly stop if not enough memory is available.

Known error messages

If running `run.sh/screen.sh` results in obscure error messages (`$(\r': command not found`), it might be due to different line breaks in `run.sh/screen.sh` using windows/linux. You can fix it using:

```
sed -i 's/\r$//' *.sh
```