Libra

Nowadays, machine-learned software plays an increasingly important role in critical decision-making in our social, economic, and civic lives.

Libra is a static analyzer for certifying **fairness** of *feed-forward neural networks* used for classification of tabular data. Specifically, given a choice (e.g., driven by a causal model) of input features that are considered (directly or indirectly) sensitive to bias, a neural network is fair if the classification is not affected by different values of the chosen features.

When certification succeeds, Libra provides definite guarantees, otherwise, it describes and quantifies the biased behavior.

Libra was developed to implement and test the analysis method described in:

```
 C. Urban, M. Christakis, V. Wüstholz, F. Zhang - Perfectly Parallel Fairness Certification of Neural Networks
Contidionally accepted to appear in Proceedings of the ACM on Programming Languages (OOPSLA), 2020.
```

Getting Started

Prerequisites

- Install **Git**
- Install APRON
  - Install GMP and MPFR

Linux	Mac OS X
`sudo apt-get install libgmp-dev`	`brew install gmp`
	`ln -s /usr/local/Cellar/gmp/ /usr/local/`
`sudo apt-get install libmpfr-dev`	`brew install mpfr`
	`ln -s /usr/local/Cellar/mpfr /usr/local/`

  - Install **APRON**

Linux or Mac OS X
`git clone https://github.com/antoinemine/apron.git`
`cd apron`
`./configure -no-cxx -no-java -no-ocaml -no-ppl`
`make`
`sudo make install`

- Install Python 3.7
- Install `virtualenv`:

Linux or Mac OS X
`python3.7 -m pip install virtualenv`

Installation

- Create a virtual Python environment:

Linux or Mac OS X
`virtualenv --python=python3.7 <env>`

- Install Libra in the virtual environment:

  - Installation from local file system folder (e.g., obtained with `git clone https://github.com/caterinaurban/Libra.git`):

Linux or Mac OS X
`./<env>/bin/pip install <path to Libra's folder>`

  or, alternatively:

  - Installation from GitHub:

Linux or Mac OS X
`./<env>/bin/pip install git+https://github.com/caterinaurban/Libra.git`

  A specific commit hash can be optionally specified by appending `@<hash>` to the command.

Command Line Usage

Libra expects as input a *ReLU-based feed-forward neural network* in Python program format. This can be obtained from a Keras model using the script `keras2python.py` (within Libra's `src/libra/` folder) as follows:

Linux or Mac OS X
`python3.7 keras2python.py <model>.h5`

The script will produce the corresponding `<model>.py` file. In the file, the inputs are named `x00`, `x01`, `x02`, etc.

A *specification* of the input features is also necessary for the analysis. This has the following format, depending on whether the chosen sensitive feature for the analysis is categorical or continuous:

Categorical	Continuous
`number of inputs representing the sensitive feature`	`1`
`list of the inputs, one per line`	`value at which to split the range of the sensitive feature`

The rest of the file should specify the other (non-sensitive) categorical features. The (non-sensitive) features left unspecified are assumed to be continuous.

For instance, these are two examples of valid specification files:

Categorical	Continuous
2	1
x03	x00
x04	0.5
2	2
x00	x01
x01	x02

In the case on the left there is one unspecified non-sensitive continuous feature (`x02`).

To analyze a specific neural network run:

Linux or Mac OS X
`./<env>/bin/libra <specification> <neural-network>.py [OPTIONS]`

The following command line options are recognized:

```
 --domain [ABSTRACT DOMAIN]

     Sets the abstract domain to be used for the forward pre-analysis.
     Possible options for [ABSTRACT DOMAIN] are:
     * boxes (interval abstract domain)
     * symbolic (combination of interval abstract domain with symbolic constant propagation [Li et al. 2019])
     * deeppoly (deeppoly abstract domain [Singh et al. 2019]])
     Default: symbolic

--lower [LOWER BOUND]

     Sets the lower bound for the forward pre-analysis.
     Default: 0.25

--upper [UPPER BOUND]

     Sets the upper bound for the forward pre-analysis.
     Default: 2

--cpu [CPUs]

     Sets the number of CPUs to be used for the analysis.
     Default: the value returned by cpu_count()
```

During the analysis, Libra prints on standard output which regions of the input space are certified to be fair, which regions are found to be biased, and which regions are instead excluded from the analysis due to budget constraints.

The analysis of the running example from the paper can be run as follows (from within Libra's `src/libra/` folder):

```
  <path to env>/bin/libra tests/toy.txt tests/toy.py --domain boxes --lower 0.25 --upper 2
```

Another small example can be run as follows (again from within Libra's `src/libra/` folder):

```
  <path to env>/bin/libra tests/example.txt tests/example.py --domain boxes --lower 0.015625 --upper 4
```

The `tests/example.py` file represents a small neural network with three inputs representing two input features (one, represented by `x`, is continuous and one, represented by `y0` and `y1`, is categorical). The specification `tests/example.txt` tells the analysis to consider the categorical feature sensitive to bias. In this case the analysis should be able to certify 23.4375% of the input space, find bias in 71.875% of the input space, and leave 4.6875% of the input space unanalyzed. Changing the domain to `symbolic` or `deeppoly` should analyze the entire input space finding bias in 73.44797685362308% of it. The input regions in which bias is found are reported on standard output.

Step-by-Step Experiment Reproducibility

The experimental evaluation was conducted on a 12-core Intel ® Xeon ® X5650 CPU @ 2.67GHz machine with 48GB of memory.

RQ1: Detecting Seeded Bias

The results of the experimental evaluation performed to answer RQ1 are shown in Tables 7-9 and summarized in Table 1. To reproduce them one can use the script `german.sh` within Libra's `src/libra/` folder. This expects the full path to Libra's executable as input:

```
./german.sh <path to env>/bin/libra
```

The script will generate the corresponding log files in Libra's `src/libra/tests/german/logs`. These can be manually inspected or a table summary of them can be generated using the script `fetch.ch` in Libra's `src/libra/tests/german/logs` folder.

> Please take note of the expected execution times before launching the script. On a less powerful machine than that used for our evaluation it might be preferable to comment out the most time consuming lines from the script before launching it.

In the `src/libra/tests/german` folder are also present the original dataset `german.csv` and the artificially fair and biased datasets `german-fair.csv` and `german-bias.csv`, as well as the 8 neural networks trained on each of these datasets.

RQ2: Answering Bias Queries

The results of the experimental evaluation performed to answer RQ2 are shown in Tables 10-12 and summarized in Table 2. To reproduce them one can use the script `compas.sh` within Libra's `src/libra/` folder. This expects the full path to Libra's executable as input:

```
./compas.sh <path to env>/bin/libra
```

The script will generate the corresponding log files in Libra's `src/libra/tests/compas/logs`. These can be manually inspected or a table summary of them can be generated using the script `fetch.ch` in Libra's `src/libra/tests/compas/logs` folder.

> Please take note of the expected execution times before launching the script. On a less powerful machine than that used for our evaluation it might be preferable to comment out the most time consuming lines from the script before launching it.

In the `src/libra/tests/german` folder are also present the original dataset `compas.csv` and the artificially fair and biased datasets `compas-fair.csv` and `compas-bias.csv`, as well as the 8 neural networks trained on each of these datasets.

RQ3: Effect of Model Structure on Scalability

The results of the experimental evaluation performed to answer RQ3 are shown in Table 3. To reproduce them one can use the script `census1.sh` within Libra's `src/libra/` folder. This expects the full path to Libra's executable as input:

```
./census1.sh <path to env>/bin/libra
```

The script will generate the corresponding log files in Libra's `src/libra/tests/census/logs1`. These can be manually inspected or a table summary of them can be generated using the script `fetch.ch` in Libra's `src/libra/tests/census/logs1` folder.

> Please take note of the expected execution times before launching the script. On a less powerful machine than that used for our evaluation it might be preferable to comment out the most time consuming lines from the script before launching it.

In the `src/libra/tests/census` folder is also present the original dataset `census.csv` as well as the 5 trained neural networks (`10`, `12`, `20`, `40`, `45`).

RQ4: Effect of Analyzed Input Space on Scalability

The results of the experimental evaluation performed to answer RQ4 are shown in Table 4. To reproduce them one can use the script `census2.sh` within Libra's `src/libra/` folder. This expects the full path to Libra's executable as input:

```
./census2.sh <path to env>/bin/libra
```

The script will generate the corresponding log files in Libra's `src/libra/tests/census/logs2`. These can be manually inspected or a table summary of them can be generated using the script `fetch.ch` in Libra's `src/libra/tests/census/logs2` folder.

> Please take note of the expected execution times before launching the script. On a less powerful machine than that used for our evaluation it might be preferable to comment out the most time consuming lines from the script before launching it.

In the `src/libra/tests/census` folder is also present the original dataset `census.csv` as well as the 4 trained neural networks (`20A`, `80A`, `320A`, `1280A`).

RQ5: Scalability-vs-Precision Tradeoff

The results of the experimental evaluation performed to answer RQ5 are shown in Table 5. To reproduce them one can use the script `japanese.sh` within Libra's `src/libra/` folder. This expects the full path to Libra's executable as input:

```
./japanese.sh <path to env>/bin/libra
```

The script will generate the corresponding log files in Libra's `src/libra/tests/japanese/logs`. These can be manually inspected or a table summary of them can be generated using the script `fetch.ch` in Libra's `src/libra/tests/japanese/logs` folder.

> Please take note of the expected execution times before launching the script. On a less powerful machine than that used for our evaluation it might be preferable to comment out the most time consuming lines from the script before launching it.

In the `src/libra/tests/japanese` folder is also present the original dataset `japanese.csv` as well as the trained neural network (`20`).

RQ6: Leveraging Multiple CPUs

The results of the experimental evaluation perfomed to answer RQ6 are shown in Table 6 and 13. To reproduce them one can again use the script `japanese.sh` within Libra's `src/libra/` folder. This time passing as input an additional argument indicating the number of CPUs to be used for the analysis:

```
./japanese.sh <path to env>/bin/libra 4
```

Authors

- **Caterina Urban**, INRIA & École Normale Supérieure, Paris, France