Installation

For artifact evaluation and study replication, we highly recommend using the pre-built VirtualBox VM. This VM was built using VirtualBox version 6.1. On startup, login with the username `dnnf` and password `dnnf`.

Basic Usage

Open a terminal window in the provided virtual machine. The DNNF tool can then be run as follows:

```
$ python -m dnnf PROPERTY --network NAME PATH
```

Where `PROPERTY` is the path to the property specification, `NAME` is the name of the network used in the property specification (typically `N`), and `PATH` is the path to a DNN model in the ONNX format.

To see additional options, run:

```
$ python -m dnnf -h
```

We provide the property and network benchmarks used in our evaluation here. These benchmarks are also already included in the provided VM.

To execute DNNF on a problem in one of the benchmarks, first navigate to the desired benchmark directory in `artifacts` (i.e., `acas_benchmark`, `neurifydave_benchmark`, or `ghpr_benchmark`). Then run DNNF as specified above. For example, to run DNNF with the Projected Gradient Descent adversarial attack from cleverhans on an ACAS property and network, run the following from within the `artifacts/acas_benchmark` directory:

```
$ cd artifacts/acas_benchmark
$ python -m dnnf properties/property_2.py \
> --network N onnx/N_3_1.onnx \
> --backend cleverhans.ProjectedGradientDescent
```

Which will produce output similar to:

```
Falsifying: Forall(x0, (((x0 <= [[ 0.68 0.5  0.5  0.5 -0.45]])
& ([[ 0.6 -0.5 -0.5  0.45 -0.5 ]] <= x0)) ==> (numpy.argmax(N(x0)) != 0)))
```

```
dnnf
  result: sat
  time: 2.6067
```

Several warnings may be produced by some of DNNF's dependencies, which can be safely ignored. The `-q` option should suppress most of these warnings.

The available backends for falsification are:

- `cleverhans.LBFGS`, which also requires setting parameters `--set cleverhans.LBFGS y_target "[[-1.0, 0.0]]"`
- `cleverhans.BasicIterativeMethod`
- `cleverhans.FastGradientMethod`
- `cleverhans.DeepFool`, which also requires setting parameters `--set cleverhans.DeepFool nb_candidate 2`
- `cleverhans.ProjectedGradientDescent`

If a property uses parameters, then the parameter value can be set using `--prop.PARAMETER=VALUE`, e.g., `--prop.epsilon=1`, similar to DNNV.

Troubleshooting

If any of the tools fail to run, these steps may help to fix the issue:

- Ensure the `DNNF` virtual environment is active. From within `/home/dnnf/DNNF/`, run `. .env.d/openenv.sh`. The shell prompt should be prefixed with `(.venv)` to indicate the virtual environment is active.