0.1 Marabou coarse refinement

The marabou SMT-based tool is used to speed up queries by utilizing the reluplex algorithm. A python interface is supplied and the module is built from source to allow for customization. Whilst the source denotes an optional dependency on the 3rd party LP solver Gurobi for MILP solving, due to licensing constraints it is currently not used.

0.1.1 Building

Overall instructions can be found at https://github.com/NeuralNetworkVerification/Marabou using the latest version. This is built from source using the following set of commands:

```
cd path/to/marabou/repo/folder
mkdir build
cd build
cmake ..
```

0.1.2 Overall structure

A `MarabouSolver` object acts as a wrapper around a Marabou network, translating from the Z3 format. In general this is done through `add` similarily to a Z3 solver. The few exceptions to this require the use of specialized functions `add_safepoint` and `add_counterexample` which are necessary when dealing with constraints to output variables. After applying constraints to the network, the constraints and the internal representation of the Z3 query (sexpr format) are outputted to a log file to be compared. The satisfiability of the input query is then determined using Marabou snc mode enabled, this is not entirely necessary however it marginally speeds up query speeds. The output is returned along with a model, is it exists, after which bounds are added on the input variables to the Z3 solver if the output is satisfiable. Bounds are used rather than absolute values to account for the accumulation of rounding errors, as Marabou uses floating point format, and to account for integer values .The size of the bounds currently is an arbitrary value however this can be changed. Finally, Z3 checks the validity of the Marabou bounds and ultimately outputs whether the model is truly satisfiable or not, preserving the original functionality of the Z3 implementation. For both finding a candidate and finding counter examples constraints on the output of the network are supplied and the solver is run.

0.1.3 Current progress

Marabou and Z3 seem to have the same internal constraints applied and most of, if not all bugs, have been eliminated. The current major concerns surround edge cases of which the causes are yet to be determined. An example of this involved Marabou and Z3's disagreement of the existence of a counter example. Marabou cannot find a counter example whilst Z3 can find a counter example. When Z3's counter example is manually checked through the network the some constraints are seemingly violated. This is an ongoing issue.

When these situations do not occur roughly 0.1% of the Marabou queries are inconsistent with Z3, this does require further testing however. Lastly, the models discovered by Marabou are yet to be validated by Z3 at the end of execution - this is a on going milestone.