Improving SAT Solver Heuristics with Graph Networks and Reinforcement Learning

Vitaly Kurin¹, Saad Godil², Shimon Whiteson¹, Bryan Catanzaro²

¹ Department of Computer Science, University of Oxford, ² NVIDIA, US.
Correspondence to vitaly.kurin@cs.ox.ac.uk

Can RL improve existing heuristics?

- Boolean Satisfiability (SAT) impacts many fields of the industry and academia, e.g. formal verification, chip design, security, combinatorial optimisation.
- SAT solvers rely on heuristics elaborately crafted with a lot of trial and error by humans.
- Some of the heuristics need a warm-up period.
- A solver should always give a correct answer.
- Pre-solving phase computation is cheap (e.g. training models)
- SAT is a sequential decision problem.

Conflict-Driven Clause Learning (CDCL)

Function CDCL(initial assignments): while not solved do
var, value ← branching heuristic();
unit propagation(var, value);
build implication graph();
analyse conflicts();
end
return SAT assignments OR unSAT

SAT problem as a graph

(a) Graph representation of \((x_1 \lor x_2) \land \neg (x_1 \lor x_3)\)
(b) Graph Q-function values for setting variables to true and false respectively.

Graph-Q-SAT (GQSAT)

- GQSAT replaces VSIDS heuristic in CDCL for the first \(k\) steps while VSIDS is warming up.
- GQSAT uses DQN with a graph neural network as a function approximator.

GQSAT reduces number of decisions by 2-3X

- GQSAT improves VSIDS.
- GQSAT generalizes across problem size.
- GQSAT generalizes from SAT to unSAT.

GQSAT makes efficient decisions from step one

GQSAT generalizes to other problem structures to a lesser extent

GQSAT is data efficient

Future Work

- Investigating graph structure influence on GQSAT performance.
- Scaling to larger problems.
- Interpreting the results using the graph structure.
- From reducing iterations to speeding up.

Acknowledgements. The authors would like to thank Rajarshi Roy, Robert Kirby, Yogesh Mahajan, Alex Aiken, Mohammad Shoeybi, Rafael Valle, Sungwon Kim, Ryan Prenger and the rest of the Applied Deep Learning Research team at NVIDIA for useful discussions and feedback. The authors would also like to thank Andrew Tao and Guy Peled for providing computing support.