NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

School of Computer Science and Engineering
College of Engineering

Applying Attention Mechanisms

in Boolean Satisfiability Neural Solvers

Student: Wu Wenxuan Supervisor: Prof. Xavier Bresson

Introduction

The Boolean Satisfiability (SAT) problem is the first proven NP-Complete problem. Solving the SAT problem as efficiently as possible has become one of the most interesting questions in theoretical computer science. As performance of traditional solvers plateau, there is an emerging field of study focusing on using neural network to solve the problem. At the same time, due to the inherent graph property of SAT problems, it becomes a natural target to be studied by Graph Neural Network (GNN). In this project, the aim is to improve on existing neural solvers by using attention mechanisms and also propose new architecture to solve SAT.

What is SAT?

The Cook-Levin Theorem states that the Boolean satisfiability problem (SAT) is NP-Complete. Despite its computational "hardness", the formulation of the problem is easy to understand. Given a Boolean formula of the form:

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$$

The Boolean satisfiability problem asks whether there is a truth value assignment of Boolean variables, such that the Boolean formula evaluates to true. The above formula is satisfiable.

Some Interesting SAT Properties

Permutation and Negation Invariance

Satisfiability is not changed if the order of clause is changed, or if the order of literals within a clause is changed, or if variables are swapped. Hence, the following formulae are equisatisfiable:

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3)$$
$$(\neg x_1 \vee \neg x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \neg x_3)$$
$$(\neg x_3 \vee x_2 \vee x_1) \wedge (x_3 \vee \neg x_2 \vee \neg x_1)$$
$$(x_3 \vee x_2 \vee \neg x_1) \wedge (\neg x_3 \vee \neg x_2 \vee x_1)$$

Self-reducibility

The solution to the decision problem of "whether the solution exists" can be used to construct the search problem of "what is the solution". This can be done in linear time.

Phase Transition

Not all SAT problem are equally hard. The "hardness" of a SAT problem is measured by **clause-to-variable** ratio. The "hardest" problem occur around **phase transition**.

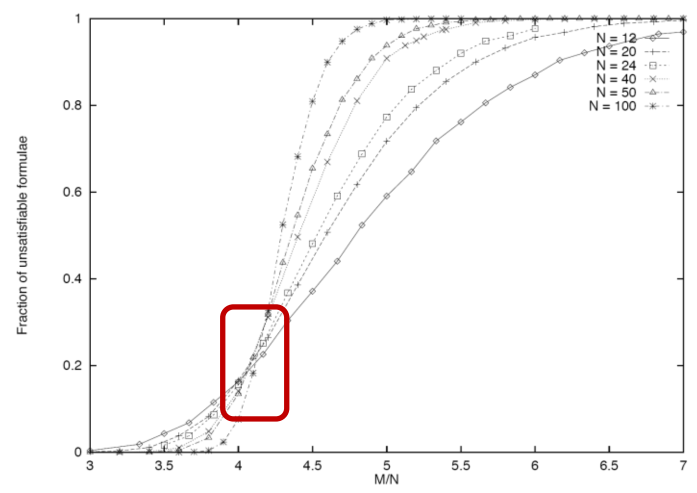

Figure 1: As the clause-to-variable ratio increases, the faction of satisfiable problem increase drastically.

Graph Neural Networks and Attention Mechanism

Graph Neural Networks (GNN) 101

Graph neural networks enable deep learning to work with non-Euclidean data structure where data cannot be effectively modelled by rows and columns.

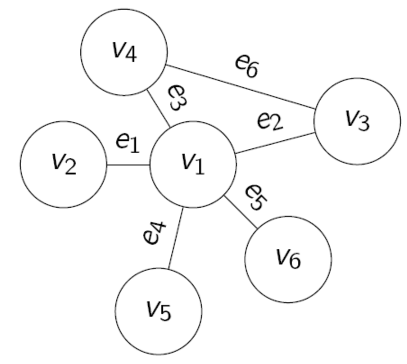

Figure 2: Example of a graph with 6 nodes and 6 edges

Attention Mechanisms 101

Originally introduced in the neural machine translation community, attention mechanism allows model to focus more on important parts of inputs by "paying attention" through assigning higher weights.

GNN + Attention

Interestingly, applications of attention mechanisms spread into the graph neural network community, resulting an interesting Graph Attention Networks (GAT) that combines the power of both.

Experiments and Results

Experiments

Modified based upon the NeuroSAT architecture, GAT layer are added in the literal and clause node embeddings update steps:

$$(C^{(t+1)}, C_h^{(t+1)}) \leftarrow \mathbf{C}_u([C_h^{(t)}, \mathrm{GAT}_f(\mathbf{L}_{\mathrm{msg}}(L_{v_j}^{(t)}))])$$

$$(L^{(t+1)}, L_h^{(t+1)}) \leftarrow \mathbf{L}_u([L_h^{(t)}, \mathrm{Flip}(L^{(t)}), \mathrm{GAT}_b(\mathbf{C}_{\mathrm{msg}}(C_{v_j}^{(t+1)}))])$$

By teaching the model to learn to attend to different neighborhood nodes by giving them different weights, one can hopefully improve the existing NeuroSAT model.

Results

Model is unable to converge, and the addition of graph attentional layers cause the existing LSTM cells in the model to unable to learn useful weights.

Recommendations and Conclusion

While there isn't many meaningful results for many experiments, there are few useful recommendations to be proposed.

- Explore the heterogeneity of SAT graphs
- Incorporate invariance properties in model design
- Individual variable classification
- Use problem-specific sat instances
- Neural solvers to guide traditional solvers

With careful and more elaborate model design, SAT problems can be solved more efficiently