Resolución de Problemas 3SAT Utilizando Redes Neuronales

Carlos Kavka y Patricia Roggero

Línea de Redes Neuronales
Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)
Departamento de Informática
Universidad Nacional de San Luis – Ejército de los Andes 950
(5700) – San Luis – Argentina

Abstract

El problema de satisfactibilidad (SAT) es un problema complejo y su estudio implica considerar instancias grandes de dicho problema. Pero es precisamente este hecho el que lo hace atractivo y ayuda a comprender la complejidad del problema. Se han realizado muchos desarrollos en distintas áreas con el fin de obtener buenos resultados para este problema. En este trabajo se propone una aproximación que usa Redes Neuronales para dar solución a algunas instancias de este problema tan particular e interesante. Las Redes Neuronales tienen una gran capacidad para aprender a resolver problemas complejos y poseen la característica de poder generalizar, puntos importantes en la resolución de instancias de SAT.

1. Introducción

Existen problemas que no pueden ser resueltos por una computadora, dado que las computadoras solamente pueden ejecutar algoritmos, esto es secuencia de instrucciones universalmente precisas y entendibles que resuelven cualquier instancia de problemas computacionales definidos rigurosamente. No es una sorpresa que esta idea intuitiva de algoritmo pueda ser definida formalmente. El correspondiente modelo matemático se llama máquina de Turing (Alan Turing, 1936).
La teoría de computabilidad tiene como objetivo el estudio de problemas de decisión, con el fin de determinar si los mismos son teóricamente decidibles.
Los problemas se pueden clasificar desde el punto de vista de la teoría de computabilidad en resolubles y no resolubles. Los problemas resolubles son objeto de estudio de la teoría de complejidad computacional.
En el contexto de complejidad computacional, el interés está centrado en establecer una medida de la cantidad de recursos computacionales (en términos de tiempo y/o espacio) necesarios para resolver un determinado problema o equivalentemente reconocer un lenguaje.
Los problemas resolubles se subdividen en tratables e intratables. Los problemas tratables son aquellos para los cuales existe un algoritmo eficiente que los resuelve. Los intratables son aquellos para los cuales no se conoce (o tal vez no exista) un algoritmo eficiente que los resuelva.

Definición: Un lenguaje L es decidible en tiempo polinomial si existe una Máquina de Turing estándard M, tal que M acepta L con $tc_M = O(n^r)$ con r un número natural independiente de n.
Definición: La familia de lenguajes decidibles en tiempo polinomial por una máquina de Turing estándard es denominada ***P***.
Definición: La familia de lenguajes aceptados en tiempo polinomial no-determinístico se denomina ***NP***.
Definición: Un lenguaje L se llama ***NP-duro*** si para todo $Q \in$ ***NP***, Q es reducible a L en tiempo polinomial. Un lenguaje ***NP-duro*** que está también en ***NP*** se llama ***NP-completo***.

2. Problema SAT

El problema SAT tiene un rol importante en la teoría de complejidad dado que fue el primero, y uno de los más simples, para el cual se demostró su condición de NP-completo.

El problema de satisfactibildad consiste en determinar si existe al menos una asignación de verdad que haga verdadera a una fórmula booleana en forma normal conjuntiva. En particular, el problema se denomina 3SAT cuando las cláusulas tienen longitud tres. Dado que cualquier fórmula en forma normal conjuntiva puede ser escrita con cláusulas de longitud tres, resolver el problema 3SAT es equivalente a resolver el problema SAT.

3. Procedimiento DPLL

Un método simple para testear satisfactibilidad, consiste en generar todos los posibles resolventes, y luego chequear si la cláusula vacía ha sido generada. Davis y Putnam introdujeron un método (llamado DP60), en el cual las variables son eliminadas una por una de la fórmula. En cada paso se generan todos los posibles resolventes considerando una variable elegida y se eliminan todas las cláusulas en las que dicha variable aparece. Cada paso genera un subproblema con una variable menos, pero posiblemente con una expansión cuadrática del número de cláusulas.
DP60 emplea dos heurísticas, la regla del literal puro y la regla de la cláusula unitaria. Ambas establecen que si alguna de las variables ocurre en la fórmula corriente en una cláusula de longitud uno, o aparece solamente negada o aparece solamente no negada, entonces se elige dicha variable como la próxima a ser eliminada.
Pero esta implementación genera un número no manejable de cláusulas resolventes, entonces Davis, Logemann y Loveland, reemplazaron la regla de eliminación por una regla de división, la cual hace que la selección de una variable conduzca a dos subproblemas más pequeños, uno por cada valor de verdad, en vez de un único, y posiblemente gran subproblema.
El procedimiento resultante (DPLL) es una búsqueda depth-first con backtracking a través de las asignaciones de verdad (parciales), aumentado con las heurísticas del literal puro y de la cláusula unitaria.

4. Aproximación propuesta

El objetivo del presente trabajo es la construcción de una red neuronal para resolver el problema 3SAT. A partir de un conjunto de fórmulas satisfactibles generadas aleatoriamente, se genera un conjunto de patrones que corresponden a las asignaciones óptimas de valores a variables que satisfacen las fórmulas. Luego se entrena una red neuronal con estos patrones y finalmente se evalúa el desempeño de esta red sobre otro conjunto de fórmulas satisfactibles y no satisfactibles generadas aleatoriamente.
El proceso consta de las siguientes etapas:

- Generación aleatoria de fórmulas satisfactibles en forma normal conjuntiva (CNF).
- Obtención de las asignaciones de longitud mínima que hacen verdaderas las fórmulas, a través de un método de búsqueda exhaustivo y generación en base a las asignaciones de un conjunto de patrones para el entrenamiento de la red neuronal.
- Conversión de los patrones generados en un conjunto adecuado de entrenamiento para la red neuronal, unificando aquellos patrones que para las mismas entradas producen distintas salidas.
- Entrenamiento de la red neuronal en base a los patrones obtenidos en la etapa anterior.
- Evaluación de la red neuronal utilizando fórmulas satisfactibles y no satisfactibles generadas aleatoriamente.

5. Resultados experimentales

Experimentos preliminares han permitido confirmar que esta aproximación produce resultados satisfactorios y que justifican avanzar en esta línea de investigación.
Por ejemplo, el algoritmo propuesto aplicado a un problema en el que el entrenamiento de la red se realiza con 20 fórmulas generadas aleatoriamente, de 40 cláusulas cada una y con 9 variables, permite obtener una red que al ser evaluada con 100 fórmulas generadas aleatoriamente, de las cuales 69 son satisfactibles y 31 son insatisfactibles, logra una performance del 99%. La calidad de los resultados es comparable con la obtenida con el algoritmo de búsqueda exhaustiva y con el DPLL.

6. Conclusiones

La performance obtenida nos conduce a pensar que este método propuesto es efectivo y además muy rápido, por lo que actualmente se está estudiando la posibilidad de generalización. Esto significa poder utilizar redes pequeñas para resolver grandes conjuntos de fórmulas, que a su vez contengan muchas cláusulas y muchas variables. Por lo tanto la investigación actualmente en desarrollo incluye determinar si una red entrenada con patrones obtenidos de fórmulas pequeñas y con pocas variables puede ser extendida a problemas con cualquier número de variables.

7. Bibliografía

Davis and Weyuker. *Introduction to Automata Theory, Languages and Computation*. Addison Wesley.

Papadimitriou and Steiglitz. *Combinatorial Optimization (Algorithms and Complexity)*. Prentice Hall.

Sudkamp Thomas. *Languages and Machines, an Introduction to the Theory of Computer Science*. Addison Wesley.

Ullman Jeffrey and Hopcroft John. *Introduction to Automata Theory, Languages and Computation*. Addison Wesley.

Haykin Simon. *Neural Networks, a Comprehensive Foundation. IEEE Computer Society Press*, MacMillan Publishing Company, 1994.

Chu Min Li and Anbulagan. *Heuristic Based on Unit Propagation for Satisfiability Problems*. Proceedings of the IJCAI-97, 1997.

David A. Clark et al. *Local Search and the Number of Solutions*. University of Strathclyde, Glasgow, Scotland, 1996.

Geffner Héctor. *Curso: Trends in Artificial Intelligence Planning*, dictado en el ámbito de la Universidad Nacional de San Luis.

Holger H. Hoos. *SAT-Encodings, Search Space Structure and Local Search Performance*. University of British Columbia, Vancouver, Canada, 1998

Stephen A. Cook and David G. Mitchell. *Finiding Hard Instances of the Satisfiability Problem: A Survey*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science.

Stuttgart Neural Network Simulator User Manual. Institute for Parallel and Distributed High Performance Systems, University of Stuttgart.