

Stage de recherche en informatique : Reconnaissance de programmes par réseaux de neurones

Sid TOUATI*
Enrico FORMENTI†

2021

Lieu

Laboratoire I3S et INRIA Sophia Antipolis, France.

Motivations et contexte

Un résultat fondamental en informatique (calculabilité) nous affirme que des problèmes indécidables existent, c'est-à-dire pour lesquels il n'existe aucun algorithme qui soit capable d'en fournir la solution. Pour un informaticien, cela n'est pas une bonne nouvelle. La situation est encore pire. En effet, le théorème de Rice nous dit que tous les problèmes sont indécidables sauf ceux qui sont triviaux (au sens de la calculabilité, bien sûr!).

Un exemple classique de problème indécidable – et que nous allons étudier dans le cadre de ce stage – consiste à prendre deux programmes et se demander s'ils calculent la même chose.

Que faire donc lorsqu'on veut s'attaquer à un problème indécidable? Il faut être pragmatique : Il faut accepter de faire des erreurs de temps en temps. Le programme qui essaye de résoudre un problème indécidable fonctionnerait bien pour certaines instances et calculerait un résultat incorrect pour d'autres. Il faudrait aussi veiller à que les cas des calculs incorrects soient aussi rares que possible.

Ce qu'on vient d'écrire est donc un cadre d'application idéal des réseaux de neurones qui, par nature, calculent leurs résultats à une marge d'erreur près.

*Professeur à l'université Côte d'Azur. Sid.Touati@inria.fr

†Professeur à l'université Côte d'Azur. Enrico.Formenti@univ-cotedazur.fr

Concrètement, dans ce stage nous voudrions construire un outil (grâce aux réseaux de neurones) qui devine ce que calcule un autre programme.

Ce problème de reconnaissance automatique des programmes, bien qu'indécidable, reste un enjeu fondamental en informatique, dont les applications sont nombreuses : *reverse engineering*, compilation, analyse de sécurité de code, etc. Essayer de deviner si un programme calcule telle ou telle fonction précise est une question qui se pose en pratique. Des travaux de recherche existent et ont tenté de reconnaître les programmes de manière algorithmique.

Présentation du sujet de stage de recherche

Ce stage a pour objectif de réfléchir à une autre démarche de reconnaissance de programmes, qui n'est pas intuitive. L'intelligence artificielle est un domaine classique issu des mathématiques et des sciences informatiques, qui a montré son efficacité en pratique dans des situations de reconnaissance de formes graphiques, de parole, de texte, d'interaction homme-machine, etc. Nous souhaitons travailler pour savoir si une telle approche peut servir à reconnaître des programmes cette fois-ci. Nous appliquerons des méthodes de réseaux de neurones pour commencer. Les étapes de travail sont :

1. Réfléchir à la bonne représentation pour la sémantique d'un programme : graphes (dépendances de données, SSA), texte (représentation fonctionnelle, programme impératif).
2. Générer via compilation plusieurs variantes d'un programme précis (par exemple une multiplication de matrices), et entraîner un réseau de neurones.
3. Tester si et à quel point un tel réseau de neurones peut reconnaître d'autres variantes du même programme.

Environnement de travail

Le langage de programmation du stage sera Python 3 avec l'appui de la bibliothèque TensorFlow mise à disposition gratuitement par Google. Le choix du langage cible de l'étude n'est pas encore fixé mais le langage C et Java sont les candidats privilégiés. Le choix final sera laissé à l'étudiant.

Prérequis souhaités

Le candidat idéal pour ce stage est un étudiant qui a suivi une formation académique en informatique (informatique fondamentale, réseaux de neurones, compilation).

Références

- [1] Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors. *Advances in Neural Information Processing Systems 31 : Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 2018.
- [2] Xinyun Chen, Chang Liu, and Dawn Song. Tree-to-tree neural networks for program translation. In Bengio et al. [1], pages 2552–2562.
- [3] D. S. Katz, J. Ruchti, and E. Schulte. Using recurrent neural networks for decompilation. In *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 346–356, 2018.