

Machine Learning : apprendre/approximer un automate à partir de ses traces

Sujet : Reconstruire un système dynamique à partir de ses traces d'exécution est un sujet ancien. En théorie des systèmes (linéaires), on parle de *system identification*, et le sujet est largement couvert essentiellement par les communautés automatique et traitement du signal. En informatique, il s'agit de retrouver un automate à partir de mots de son langage et de contre-exemples. Le problème a été initialement étudié par Dana Angluin, laquelle a proposé le célèbre algorithme L-star. Celui-ci suppose l'accès à un oracle permettant de tester l'appartenance de mots ou l'exactitude d'un modèle : si l'automate soumis ne produit pas le même langage que l'automate à deviner, l'oracle produit alors un mot contre-exemple. De nombreuses variantes existent, selon la puissance de l'oracle, voire son absence lorsque les données d'apprentissage sont figées. L'identification de systèmes dynamiques est aussi abordée par la communauté automatique pour des modèles à temps et état continus. Il s'agit alors de retrouver (ou de découvrir) des lois d'évolution du système, sous forme d'équations différentielles. De nouvelles techniques de machine learning ont été développées dans ce but. On parle d'*equation discovery*.

L'objectif de ce projet est d'explorer l'intérêt des nouvelles techniques de *machine learning* pour aborder le problème de Dana Angluin, c'est à dire identifier (ou approximer) un automate à partir d'un ensemble de traces éventuellement grand, mais donné *a priori*. Il a été démontré que les techniques à base de RNN (Recursive Neural Networks) peuvent apprendre à reconnaître un langage régulier. Étonnamment, l'espace d'état de ces systèmes, bien que vivant dans \mathbb{R}^d et bâti par des techniques de gradient stochastique, fait apparaître des *clusters* de valeurs qui reflètent d'assez près le comportement d'une machine discrète, un automate. On retrouve ainsi une propriété déjà manifeste dans les grands modèles de langage (à la GPT4), à savoir la possibilité d'approcher des dynamiques discrètes très complexes.

L'objectif de ce projet est d'explorer les capacités d'apprentissage de modèles comme les RNN, les LSTM (Long-Short Term Memory) mono- et bi-directionnels, autant que les *transformers*, dans le but d'apprendre ou d'approximer un automate caché, décrit à partir de ses traces. Plusieurs directions de recherche sont envisageables, selon le goût et les aptitudes du candidat.

Les premiers travaux pourront par exemple explorer des problèmes de type classification binaire de mots (i.e. reconnaissance de langage formel), afin d'analyser les critères sur lesquels se bâtit une représentation d'état dans \mathbb{R}^d , notamment pour forcer la concentration d'état et éviter la duplication inutile (i.e. aller vers l'apprentissage d'un automate minimal). On pourra aussi examiner la façon dont ces approches peuvent s'étendre en des modèles génératifs, en entraînant le modèle à prédire la prochaine lettre d'un mot. Dans ces deux cas, une question essentielle reste l'estimation d'une distance entre le langage appris et le vrai langage caché révélé via un ensemble fini de traces, c'est à dire la capacité de généralisation du modèle. On tentera de proposer plusieurs manières de formaliser cette question.

Une autre direction de recherche porte sur l'utilisation d'auto-encodeurs variationnels pour comprimer un ensemble de traces vers un domaine de représentation compact (le *feature space*), suffisant pour reconstruire les traces de la base de données. Certains travaux montrent qu'il est alors possible d'utiliser l'espace compact de représentation comme un espace d'état, sur lequel identifier la dynamique du système caché. Approche suivie par exemple en *equation discovery* par S. Brunton *et al.* On pourra explorer l'intérêt de cette approche pour des automates finis.

Sur l'un des axes proposés, on souhaite que le candidat soit en mesure de poser formellement le problème d'apprentissage pour la structure de réseau de neurones choisie, comprenne en profondeur les algorithmes nécessaires, soit en mesure de les implémenter ou d'adapter des codes existants, et enfin mette en oeuvre un plan d'expérience pour les tester. Le compromis entre travail expérimental et développements théoriques s'adaptera aux centres d'intérêts du candidat.

Bibliographie :

- Dana Angluin's L* algorithm : "Learning Regular Sets from Queries and Counter-Examples," 1987, doi :10.1016/0890-5401(87)90052-6
- J. Michalenko, A. Shah, A. Verma, R. Baraniuk, S. Chaudhuri, A. Patel, "Representing Formal Languages : A Comparison Between Finite Automata and Recurrent Neural Networks," ICLR 2019.
- W. Merrill, "Formal Language Theory Meets Modern NLP," 2021, arXiv :2102.10094
- Diederik Kingma, Max Welling, "An introduction to variational autoencoders," arXiv :1906.02691
- Equation Discovery, by Steven Brunton : <https://www.youtube.com/watch?v=KmQkDgu-Qp0>

Contacts : Eric Fabre (eric.fabre@inria.fr)